NSN-0056-00-471-9000 TI 6490.19

INSTRUCTION BOOK

TANDEM PROCESSING SUBSYSTEM

(DIAGNOSTICS)

PART OF

FLIGHT SEVICE AUTOMATION SYSTEM

VOLUME II

CONTRACT DTFA01-81-C-10039

CONTRACTOR

E-SYSTEMS, INC.
GARLAND DIVISION
P.O. BOX 660023
DALLAS, TEXAS 75266-0023

416-21626

3 JUNE 1985

MADE FOR

U.S. DEPARTMENT OF TRANSPORTATION

FEDERAL AVIATION ADMINISTRATION

PROPRIETARY NOTICE

Tandem will grant to the FAA, with regard to Tandem's proprietary software and hardware manuals, including but not limited to maintenance manuals, Limited Rights in Data in accordance with DAR 7-104.9(a), with the following exceptions:

1. "Government" is limited to "FAA FSAS and MPS Programs only."

2. No right to reproduce is granted.

3. No right to disclose to third parties is granted unless either (a) the third party is authorized by Tandem, in writing, to receive such disclosure; or (b) Tandem has declined, in writing, to assist the FAA in resolving the emergency repair or overhaul work. In no event will the FAA make such disclosure without making such disclosure subject to a suitable prohibition against further use, release, or disclosure.

The rights identified above shall be subject to and in accordance with the Tandem/FAA License Agreement executed by the FAA on 6 December 1984.

VOLUME 2

Serial Number

238

TANDEM

NonStop II™

Diagnostic Operating Procedures

System Maintenance Manual

Part No. 82803

CURRENT ORGANIZATION OF THE NonStop II DIAGNOSTICS MANUAL

The numbering scheme for the DIAGNOSTIC OPERATING PROCEDURES manual, Part Number 82803, was initially planned to accommodate all SHADOW diagnostics, those already in the field, those under development, and those that were proposed as possible future projects. As a result, this manual has gaps in the numbering sequence of its various volumes, chapters, parts, and sections.

The cover sheet of the manual, NOTE TO THE USER, gives a complete explanation of the DIAGNOSTIC OPERATING PROCEDURES manual numbering plan and references volumes of the manual that have not been written.

The Chapters and Parts that are included in the present manual are:

Volume 1, Chapter 1, Part 6A	GENERAL TEST
Volume 1, Chapter 1, Part 6B	MEMORY DIAGNOSTIC
Volume 1, Chapter 1, Part 6C	LOADABLE CONTROL STORE
Volume 1, Chapter 2, Part 1	LOADING SHADOW AND OTHER STANDALONE
Volume 1, Chapter 2, Part 2	TAKING A PROCESSOR MODULE OFF LINE
Volume 1, Chapter 2, Part 3	RELOADING A PROCESSOR MODULE
Volume 1, Chapter 2, Part 6	USING EXERCISE
Volume 1, Chapter 3, Part 1	DIAGBOOT
Volume 2, Chapter 2, Part 1	BUS16
Volume 3B, Chapter 1, Part 1	OVERVIEW: OFFLINE DISC DIAGNOSTICS
Volume 3B, Chapter 1, Part 1A	FAST
Volume 3B, Chapter 1, Part 1B	DISC
Volume 3B, Chapter 1, Part 1C	DEX
Volume 3B, Chapter 1, Part 1D	DUT
Volume 3B, Chapter 1, Part 1E	EDT
Volume 3B, Chapter 1, Part 1F	FMT
Volume 3B, Chapter 1, Part 1G	REX

MINOR FORMAT CHANGES MADE IN SECOND PRINTING OF 82803 DIAGNOSTICS

To help the reader use the NonStop II DIAGNOSTIC OPERATING PROCEDURES manual, several changes have been made:

1. All the available chapters have been collated into one package

2. Tab dividers have been added

3. Bleeder tabs have been eliminated

4. The General Table of Contents and the Block Diagram that showed the proposed eventual structure of the manual have been removed.

NOTE TO THE USER

The purpose of this note is to explain the overall structure of the diagnostic manuals and thereby facilitate each user's finding the information needed in any given circumstances.

There are two diagnostic manuals presently, one documenting the NonStop System (NonStop Diagnostic Operating Procedures, Part Number 82802), and the other documenting the NonStop II System (NonStop II Diagnostic Operating Procedures, Part Number 82803). Each diagnostic manual was written as a reference manual. Because each manual must accommodate many different users, who have a variety of needs and a range of experience with Tandem hardware and software, each necessarily contains information that will not be required by every user in every circumstance.

The manual is divided, first of all, into five volumes. The contents of these volumes parallel discernible levels of hardware in the Tandem system. Volume 1 describes system-level diagnostics, and Volume 2, processor/bus-level diagnostics. Volume 3A deals with communication subsystem diagnostics; Volume 3B, with disc drive subsystem diagnostics; and Volume 3C, with tape drive subsystem diagnostics. The user who wishes to run bus diagnostics locates himself at once in Volume 2, while the user testing disc drives selects Volume 3B, and so on.

Each volume is in turn divided into chapters, parts, and, where necessary, subparts. The nature of these divisions is determined primarily by the nature of the material covered. In Volumes 3A through 3C, for instance, the chapters distinguish Tandem-produced from vendor-produced diagnostics; within chapters, the parts distinguish offline from online diagnostics; and within parts, the subparts deal with individual diagnostics. At this level, the presentation characteristically falls into five numbered sections: Introduction, Overview, Commands, Error Conditions and Dispositions, and Run Procedure.

Section 1 identifies by name, release number, and revision level the diagnostic being described. It lists the devices to which the diagnostic applies. If appropriate, it may place the diagnostic in the larger context of other Tandem diagnostics available for the same devices. It may also briefly characterize the nature of the diagnostic or of the system environment in which it runs. Where pertinent, it states the hardware and software revision levels with which the diagnostic is compatible.

Section 2 describes the diagnostic in general terms and sketches the outlines of diagnostic execution, using flowcharts or other diagrams where feasible. It lists the individual tests or subtests performed, as well as any run-time options selectable at initialization. It also gives an estimate of average run times.

Section 3 lists in tabular form any program-defined commands available in the diagnostic, including mnemonic, parameters, and defaults. It then gives detailed descriptions of each command, followed by examples. If command execution results in one or more screen displays, these displays are depicted.

Section 4 describes the error messages that may be returned by the diagnostic and, so far as possible, provides contexts for these error messages. It also provides whatever hardware-related information is needed for interpretation of error messages, such as data formats, device specifications, and the contents of the EIO and IIO status and cause words. Where possible, error messages are tabularized for ease of identification.

Section 5 summarizes the preparations for running the diagnostic, including whatever must be done to ready the system environment and the specific hardware to be tested. It then gives a step-by-step description of the actual run procedure. This includes all program prompts for input and the possible user responses. Where pertinent, steps are prefaced with warnings or cautions. Other information useful at a given step, but not vital to program execution, may follow the step in a note. Section 5 closes, where feasible, with a flowchart of the run procedure including steps to be taken in the event of run procedure failures.

The user who is unfamiliar with a given diagnostic may wish to read the entire five-section description of that diagnostic, while the user who needs only to determine the significance of a particular error message turns directly to the tables in Section 4. Similarly, the user who needs only to refresh his memory of the run procedure simply skims the numbered steps of Section 5, ignoring the notes and other amplifying material.

VOLUME 1

System Diagnostics, Procedures and Utility Programs

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1

SYSTEM DIAGNOSTICS, PROCEDURES, AND UTILITY PROGRAMS

(CURRENT)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

CURRENT TABLE OF CONTENTS

VOLUME 1 – SYSTEM DIAGNOSTICS, PROCEDURES, AND UTILITY PROGRAMS

CHAPTER 1 – SYSTEM DIAGNOSTICS

PART 6A. GENERAL TEST..

PART 6B. MEMORY DIAGNOSTIC....................................

PART 6C. LOADABLE CONTROL STORE...............................

CHAPTER 2 – DIAGNOSTIC PROCEDURES

PART 1. LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS..............

PART 2. TAKING A PROCESSOR MODULE OFF LINE.........................

PART 3. RELOADING PROCESSOR MODULES................................

PART 6. USING EXERCISE...

CURRENT TABLE OF CONTENTS (Cont'd)

VOLUME 1 – SYSTEM DIAGNOSTICS, PROCEDURES, AND UTILITY PROGRAMS (Cont'd)

CHAPTER 3 – DIAGNOSTIC UTILITY PROGRAMS

PART 1. DIAGBOOT...

CHAPTER 1

System Diagnostics

PART 6

Microdiagnostics

PART 6A

General Test

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 1, PART 6A

GENERAL TEST

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

REVISION RECORD

REVISION	DATE	DESCRIPTION
A	Aug. 1981	First Edition
B	Dec. 1982	Editorial changes.

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 1, CHAPTER 1, PART 6A

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION....................................1-1

SECTION 2
OVERVIEW

2.0 OVERVIEW..2-1
2.1 PROGRAM INITIALIZATION..........................2-1
2.2 PROGRAM EXECUTION...............................2-2
2.3 EXAMINATION OF ERROR MESSAGES...................2-2
2.4 REINITIALIZATION OF THE DIAGNOSTIC..............2-3

SECTION 3
COMMANDS

3.0 COMMANDS..3-1
3.1 SCREEN SELECTION AND MANIPULATION...............3-1
3.1.1 Diagnostic Control Screen.....................3-2
3.1.2 Error Message Screen..........................3-4
3.1.3 Error Log Screen..............................3-6
3.2 FUNCTION KEY COMMANDS USED IN GENERAL TEST.......3-6

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS...............4-1

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES..................................5-1
5.1 SPECIAL CONSIDERATIONS..........................5-7

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Fig.No.	Title	Page No.
3-1	Diagnostic Control Screen	3-3
3-2	Error Message Screen	3-5
3-3	Error Log Screen	3-7
5-1	Processor Status Screen	5-2
5-2	Microdiagnostic Loader Screen	5-3

LIST OF TABLES

Table No.	Title	Page No.
3.1	Screens Available in General Test	3-1
3.2	Function Key Commands Used in General Test	3-8
4.1	Error Types and Probable Areas of Failure	4-3
4.2	Interpretation of Control Panel Lights	4-6

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The Processor Functional Test (T9465A01.CPU420A), or General Test diagnostic, tests all major components of the Tandem NonStop II (TM) system to ensure that they are functioning properly. When a Tandem processor is generating errors, General Test can be run to determine the error type and to isolate the probable area of failure. After the probable failing board has been replaced, General Test can be run again as an end test to ensure that the errors have been eliminated.

General Test is a downloadable program. It resides on one of the two flexible (floppy) discs associated with the Operations and Service Processor (OSP) and is run through the OSP. General Test runs four sets of tests in the following sequence:

a. Instruction Processing Tests

b. Memory Control and Interprocessor Bus Tests

c. Control Store and I/O Channel Tests

d. Instruction Cycling Tests

The approximate run time for one full pass through these tests is five minutes. General Test is designed to run on all NonStop II Operating Systems beginning with revision A00.

Operator interface to General Test occurs through one of three screens generated by the special message handler for this diagnostic. They are displayed at the OSP terminal and manipulated by the use of predefined function keys. These screens are

a. Diagnostic Control Screen

b. Error Message Screen

c. Error Log Screen

For a complete description of the OSP, its functions, and its specific relation to the downloadable diagnostics, refer to Tandem OSP User Guide, 82801-B00.

PART 6A - GENERAL TEST
INTRODUCTION

SECTION 2
OVERVIEW

2.0 OVERVIEW

The OSP must be functioning properly in order to run General Test. Further, for all parts of General Test to run successfully, certain PROM-resident microroutines must also be in place and operating correctly.

The processor to be tested must first be selected on the Processor Status Screen of the OSP. General Test is then accessed through the Microdiagnostic Loader Screen of the OSP.

Each of the four test sets is designed to test functions associated with a particular board. The Instruction Processing Tests test the Instruction Processing Unit (IPU). The Memory Control and Interprocessor Bus Tests test the Memory Control Board (MCB). The Control Store and I/O Channel Tests test the Channel/Control Store/ Diagnostic Data Transceiver board (CCD). The Instruction Cycling Tests test both IPU and MCB functions.

User control of General Test is effected through three special displays or screens. (Refer to Table 3-1.) Each screen is composed of various control fields. (Refer to paragraphs 3.1.1 through 3.1.3.) Modification of the control fields within a screen gives the user access to the control functions of General Test.

2.1 PROGRAM INITIALIZATION

Program initialization is instantaneous. When General Test is selected from the diagnostics available on the Microdiagnostic Loader Screen, the microcode for its special message handler is immediately downloaded from floppy disc into OSP RAM. The message handler then performs the following functions:

a. Sets up the operating environment.

b. Defaults to the Diagnostic Control Screen.

c. Enters an idle loop and waits for operator instructions.

2.2 PROGRAM EXECUTION

The Diagnostic Control Screen automatically displays information about processor configuration, status of diagnostic execution, and detection of errors in the processor under test. This screen also enables the modification of three run-time parameters: Halt on Error, Loop on Test, and Diagnostic Test Number.

When the fields are modified as desired and the START TEST function key (F9) is depressed, the message handler reads from floppy disc and loads into Loadable Control Store (LCS) of the processor under test the microcode for the selected set of tests. When execution is complete, the message handler reruns the selected set of tests if Loop on Test is enabled; otherwise, it enters an idle loop and waits for further instructions. If the option to run all four sets of tests is selected, the message handler first loads and runs the IPU tests. When the IPU tests complete, the message handler loads and runs the next set of tests, and so on until all four sets of tests run. Then, again depending on the setting of Loop on Test, the message handler either repeats the procedure, starting with the first set of tests, or enters an idle loop and waits for operator instructions.

If an error is detected during execution of a test, an error handler routine in the diagnostic responds by sending an error message to the message handler. Errors are logged in a table maintained by the message handler, and information about them is displayed on the Error Message Screen and Error Log Screen.

The status of General Test after transmission of an error message depends on the setting of the Halt on Error control field. If Halt on Error is disabled, execution of the test resumes at the point of error detection. If Halt on Error is set, the error handler passes control to the command interpreter and suspends the diagnostic.

2.3 EXAMINATION OF ERROR MESSAGES

While General Test is executing, the Error Message Screen and Error Log Screen can be examined for information about errors being encountered. These screens may be examined while a diagnostic test is running, or the diagnostic test may first be suspended. This choice is entered from the Diagnostic Control Screen.

2.4 REINITIALIZATION OF THE DIAGNOSTIC

General Test can be reinitialized at any time from any of the three screens by depressing the appropriate function key. Reinitialization terminates the diagnostic being run, clears the Error Message Screen and Error Log Screen of accumulated information, redisplays the Diagnostic Control Screen if in some other screen, returns the run-time parameters and passcount to the default settings, and places the message handler in an idle loop, where it waits for operator instructions.

PART 6A - GENERAL TEST OVERVIEW

B00

SECTION 3
COMMANDS

3.0 COMMANDS

The NonStop II General Test is accessed and controlled through preformatted screens and predefined function keys on the OSP. Operator interface does not take place directly through a command interpreter but through a diagnostic message handler that generates the screens and defines the various function keys needed to manipulate them.

3.1 SCREEN SELECTION AND MANIPULATION

At startup, the message handler for General Test defaults to the Diagnostic Control Screen. (Refer to paragraph 3.1.1.) After this point, any one of the three available screens can be fetched from any other by depressing the function key associated with the desired screen. From each of the given screens, the necessary test operations can then be performed by other function keys defined in the message handler.

A special function key (F8) terminates General Test and returns control to the Processor Status Screen. For a description of the use of this function key command when General Test is running in more than one processor at the same time, refer to paragraph 5.1. For further information, refer to paragraph 3.3.5 of the OSP User Guide, 82801-B00. The screens and their associated function keys are listed in Table 3.1. The formats of the General Test screens and the specific function key commands executable from each are described in paragraphs 3.1.1 through 3.1.3.

Table 3.1 Screens Available in General Test

FUNCTION KEY	SCREEN DISPLAYED
F1	Diagnostic Control Screen
F2	Error Message Screen
F3	Error Log Screen

3.1.1 Diagnostic Control Screen

The Diagnostic Control Screen is invoked by depressing function key F1. The format of the Diagnostic Control Screen is shown in Figure 3-1.

At the top of the Diagnostic Control Screen is a banner identifying the processor number, test name, and release date. At the bottom is a summary of the relevant function keys and their uses. The remainder of the screen is divided into three active fields: Diagnostic Control, Diagnostic Test Number List, and Processor Configuration.

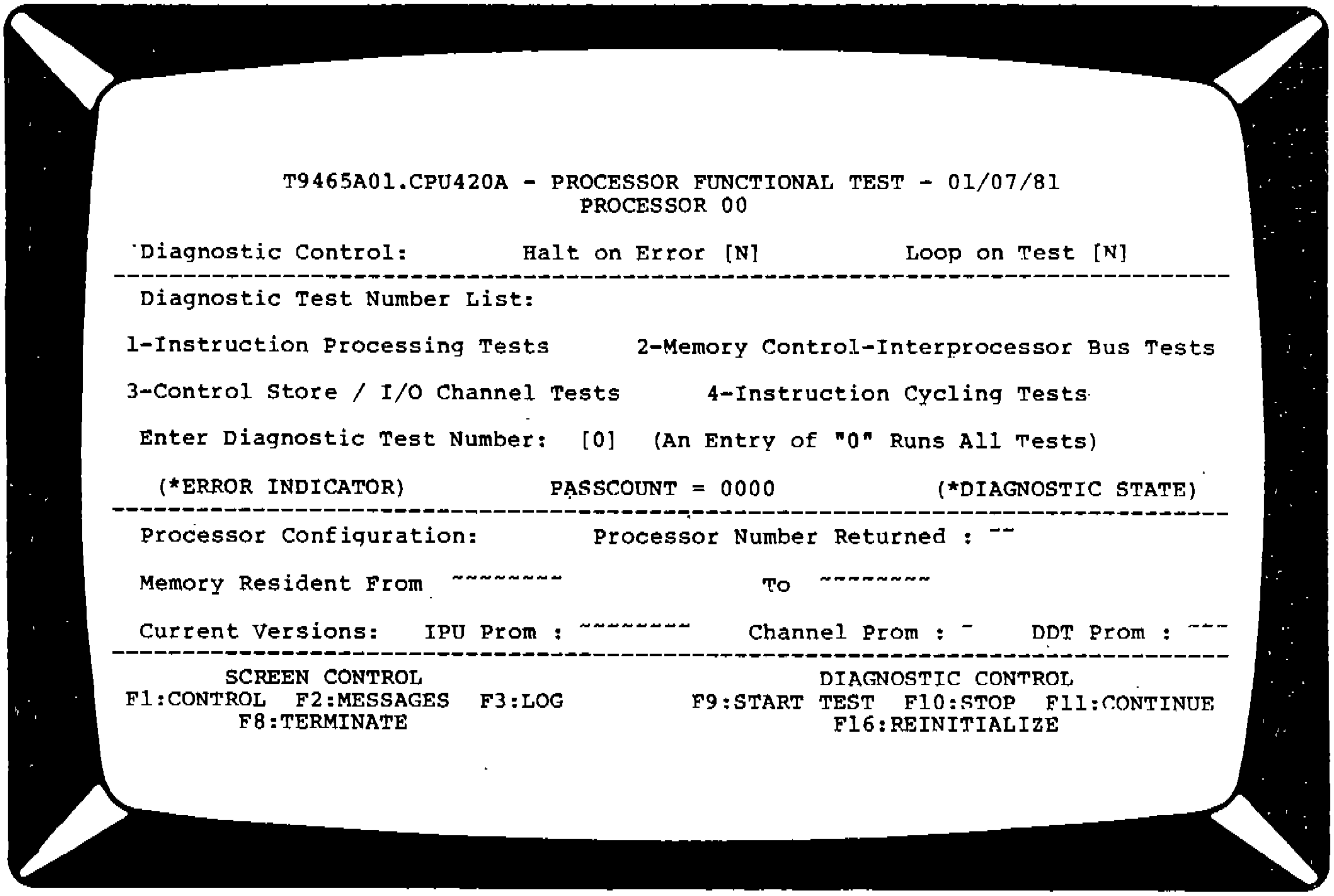
The field of the Diagnostic Control Screen identified as Diagnostic Control contains two parameters which can be altered as dictated by the needs of the situation. The default settings are Halt on Error set to N (disabled) and Loop on Test set to N (disabled). If Halt on Error is enabled, the diagnostic suspends execution on encountering an error. If Loop on Test is enabled, General Test restarts itself automatically and increments PASSCOUNT by one every time the test sequence is completed.

The field of the Diagnostic Control Screen identified as Diagnostic Test Number List enumerates the four sets of tests that can be selected and provides a Test Number field where the number of the test to be run must be indicated. An entry of zero (0) in the Test Number field causes all four sets of tests to run in sequence. This area of the control screen also provides ERROR INDICATOR, DIAGNOSTIC STATE, and PASSCOUNT fields. The default setting of PASSCOUNT is zero (0).

ERROR INDICATOR and DIAGNOSTIC STATE are automatically updated by the message handler as the diagnostic runs. When General Test is invoked from the Execute Diagnostics Screen and the Diagnostic Control Screen is first displayed, ERROR INDICATOR displays NO ERRORS DETECTED, and DIAGNOSTIC STATE displays DIAGNOSTIC IDLE. When function key F9 is depressed to load the selected set of tests, DIAGNOSTIC STATE flashes DIAGNOSTIC LOADING. When the tests begin to execute, DIAGNOSTIC STATE changes to flash DIAGNOSTIC EXECUTING.

If all four sets of tests are selected for execution, DIAGNOSTIC STATE again flashes DIAGNOSTIC LOADING each time a test set completes and the message handler downloads the microcode for the next test set. As each is loaded and begins to run, DIAGNOSTIC STATE returns to DIAGNOSTIC EXECUTING. DIAGNOSTIC STATE thus alternates between DIAGNOSTIC LOADING and DIAGNOSTIC EXECUTING as the test sequence proceeds. The particular set of tests currently executing is always highlighted by inverse video in the Diagnostic Test Number List.

The state of General Test on completion of a selected test is governed by the setting of Loop on Test. If Loop on Test is enabled, PASSCOUNT increments and the test restarts. If Loop on Test is disabled, DIAGNOSTIC STATE changes from DIAGNOSTIC EXECUTING to DIAGNOSTIC IDLE, and the program waits for further instructions.



```
           T9465A01.CPU420A - PROCESSOR FUNCTIONAL TEST - 01/07/81
                              PROCESSOR 00

 Diagnostic Control:       Halt on Error [N]         Loop on Test [N]
--------------------------------------------------------------------------
 Diagnostic Test Number List:

1-Instruction Processing Tests      2-Memory Control-Interprocessor Bus Tests

3-Control Store / I/O Channel Tests       4-Instruction Cycling Tests

 Enter Diagnostic Test Number:  [0]  (An Entry of "0" Runs All Tests)

  (*ERROR INDICATOR)          PASSCOUNT = 0000          (*DIAGNOSTIC STATE)
--------------------------------------------------------------------------
 Processor Configuration:        Processor Number Returned : ~~

 Memory Resident From  ~~~~~~~~             To  ~~~~~~~~

 Current Versions:  IPU Prom : ~~~~~~~~   Channel Prom : ~   DDT Prom : ~~~
--------------------------------------------------------------------------
         SCREEN CONTROL                           DIAGNOSTIC CONTROL
F1:CONTROL  F2:MESSAGES  F3:LOG         F9:START TEST  F10:STOP  F11:CONTINUE
       F8:TERMINATE                              F16:REINITIALIZE
```

T16/8I16A-01

Figure 3-1 Diagnostic Control Screen

If execution is manually interrupted while a diagnostic test is running (or if an error is encountered with Halt on Error enabled), DIAGNOSTIC STATE changes from either DIAGNOSTIC EXECUTING or DIAGNOSTIC LOADING to DIAGNOSTIC SUSPENDED. It changes back again when the test is resumed. If an error is detected during execution, ERROR INDICATOR flashes ERRORS DETECTED.

The field of the Diagnostic Control Screen identified as Processor Configuration contains information that is updated only when a full pass of the selected test set is completed. The information returned includes: processor number (in octal), upper and lower limits of memory, and current versions of IPU, Channel, and DDT PROM. If any noncontiguous memory is encountered when the limits of memory are being searched, ERROR INDICATOR changes from NO ERRORS DETECTED to ERRORS DETECTED.

The field at the bottom of the Diagnostic Control Screen summarizes the function keys which can be exercised from the screen. On the left are those for moving to the other available screens; on the right, those for executing the commands specific to this screen.

Depressing function key F9 starts the diagnostic running. Depressing F10 suspends a test at any time during its execution. Depressing F11 restarts a test suspended by a previous F10 command (or one which has encountered an error with HALT ON ERROR enabled). Depressing F16 reinitializes General Test.

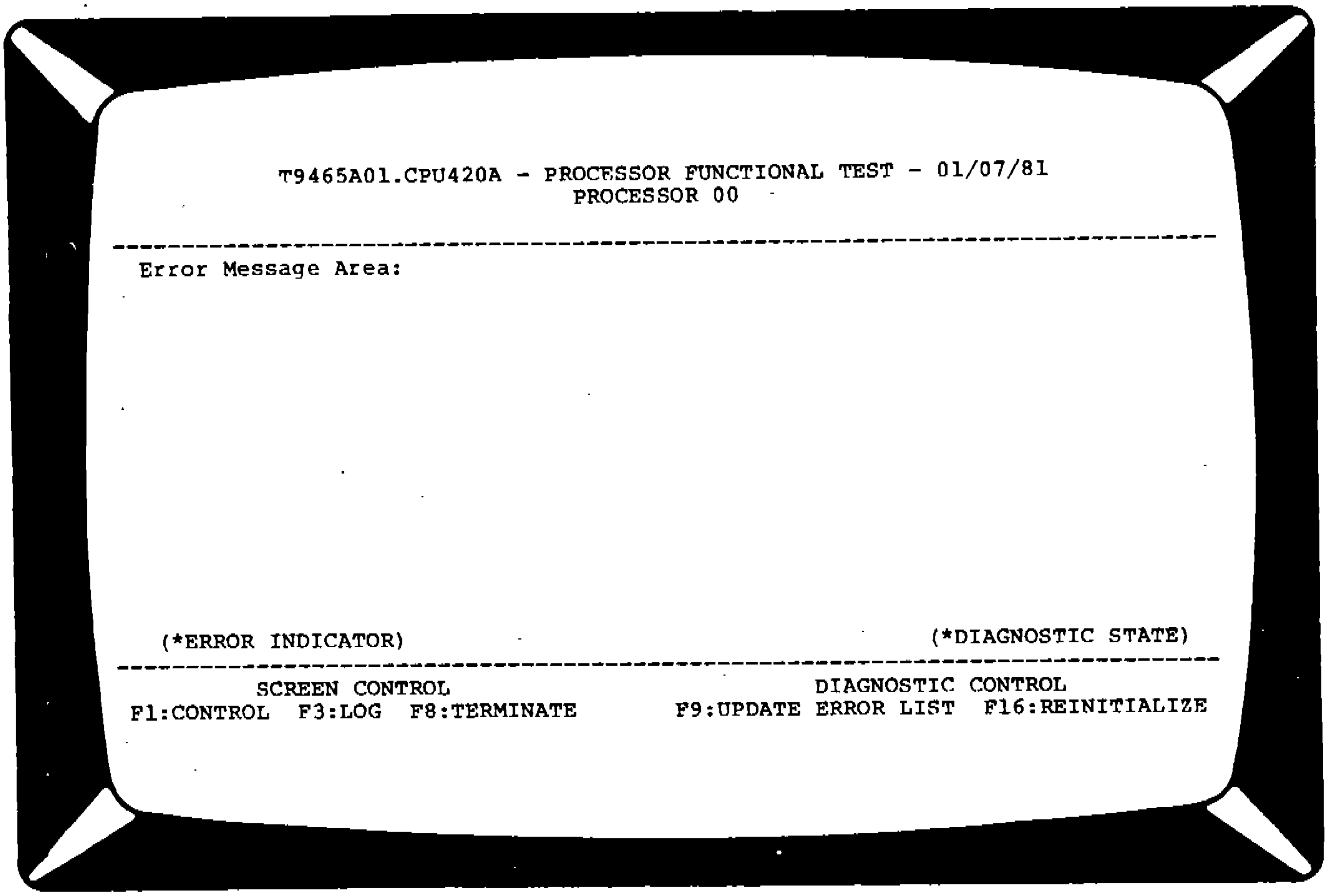
3.1.2 Error Message Screen

The Error Message Screen is invoked by depressing function key F2. The format of the Error Message Screen is shown in Figure 3-2.

At the top of the Error Message Screen is the banner identifying processor number, test name, and release date. At the bottom is the summary of function keys that can be exercised from this screen. The remainder of the screen is devoted to the Error Message Area.

Error Message Area displays information about errors encountered, including: the type of error, the test set running when the error was detected, and an octal number indexing the error to a specific subtest within the test set. This number corresponds to the octal value displayed in the control panel lights while the test runs. For further information about the individual subtests run during each test set, refer to the OSP User Guide, 8801-B00.

The maximum number of errors that can be displayed at any one time is seven. In test sequences that encounter more than seven errors, the screen retains the first six errors encountered. The seventh entry continues to change so that it always records the most recent error encountered. For details of message interpretation and fault isolation, refer to paragraph 4.0.



```
          T9465A01.CPU420A - PROCESSOR FUNCTIONAL TEST - 01/07/81
                                PROCESSOR 00
-------------------------------------------------------------------------
 Error Message Area:










  (*ERROR INDICATOR)                                (*DIAGNOSTIC STATE)
-------------------------------------------------------------------------
          SCREEN CONTROL                          DIAGNOSTIC CONTROL
F1:CONTROL  F3:LOG  F8:TERMINATE       F9:UPDATE ERROR LIST  F16:REINITIALIZE
```

T16/8116A-02

Figure 3-2 Error Message Screen

Error Message Area also contains the ERROR INDICATOR and DIAGNOSTIC STATE fields. These are automatically updated by the message handler as described in paragraph 3.1.1.

The Error Message Screen is updated automatically only at the end of every full pass of the selected test. This screen can be manually updated during a test sequence by depressing function key F9.

As on the other screens, depressing function key F16 reinitializes General Test.

3.1.3 Error Log Screen

Invoke the Error Log Screen by depressing function key F3. The format of the Error Log Screen is shown in Figure 3-3.

At the top of this screen is the banner identifying processor number, test name, and release date. At the bottom is a summary of the function keys that can be exercised from this screen. The remainder of the screen is devoted to the Error Log by Test Sequence.

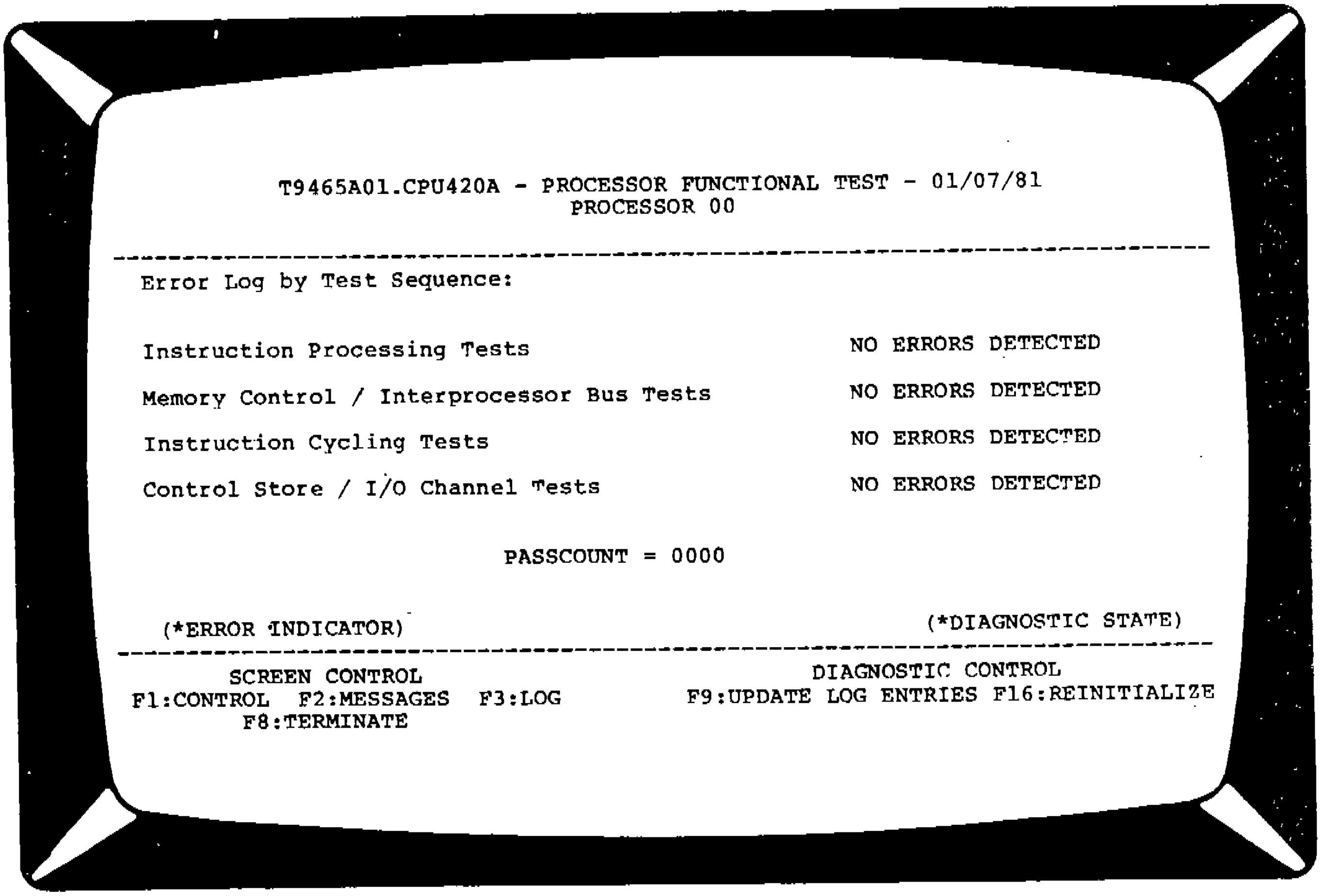
The Error Log by Test Sequence displays a running total of the errors encountered for each of the four sets of tests. It also contains the ERROR INDICATOR, DIAGNOSTIC STATE, and PASSCOUNT fields, which are automatically updated by the message handler as described in paragraph 3.1.1.

Like the Error Message Screen, the Error Log Screen is updated automatically only at the end of every full pass of a selected test but can be updated manually during a test sequence by depressing function key F9.

As on the other screens, depressing function key F16 reinitializes General Test.

3.2 FUNCTION KEY COMMANDS USED IN GENERAL TEST

A summary of all function key commands used in General Test is given in Table 3.2.



```
           T9465A01.CPU420A - PROCESSOR FUNCTIONAL TEST - 01/07/81
                               PROCESSOR 00
------------------------------------------------------------------------------
 Error Log by Test Sequence:

 Instruction Processing Tests                          NO ERRORS DETECTED

 Memory Control / Interprocessor Bus Tests             NO ERRORS DETECTED

 Instruction Cycling Tests                             NO ERRORS DETECTED

 Control Store / I/O Channel Tests                     NO ERRORS DETECTED

                            PASSCOUNT = 0000

  (*ERROR INDICATOR)                                 (*DIAGNOSTIC STATE)
------------------------------------------------------------------------------
        SCREEN CONTROL                           DIAGNOSTIC CONTROL
F1:CONTROL  F2:MESSAGES  F3:LOG         F9:UPDATE LOG ENTRIES F16:REINITIALIZE
         F8:TERMINATE
```

T16/8116A-03

Figure 3-3 Error Log Screen

Table 3.2 Function Key Commands Used in General Test

FUNCTION KEY	OPERATION PERFORMED
1	Fetch Diagnostic Control Screen
2	Fetch Error Message Screen
3	Fetch Error Log Screen
4	Not used in General Test
5	Not used in General Test
6 (shifted)	Fetch Processor Status Screen (used before General Test to select the processor to be tested)
7 (shifted)	Fetch Microdiagnostic Loader Screen for downloading General Test
8	Terminate diagnostic for the current processor; display Control Screen of any other processor running diagnostic (if none, display Processor Status Screen)
9	a. Start General Test (from the Diagnostic Control Screen) b. Update error list (from the Error Message Screen) c. Update error log (from the Error Log Screen)
10	Suspend execution of General Test (from the Diagnostic Control Screen)
11	Restart a suspended test (from the Diagnostic Control Screen)
12 through 15	Not used in General Test
16	Reinitialize General Test (from the Control, Error Message, or Error Log Screens)

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS

Information about errors detected by General Test is displayed on the Error Message and Error Log screens. These screens enable the monitoring and analysis of test results and aid in the isolation of failures to the board level. Refer to paragraphs 3.1.2 and 3.1.3 for details about the format of these screens.

When an error is encountered, the ERROR INDICATOR field of all screens immediately changes to read ERRORS DETECTED. The problem can then be analyzed by fetching the Error Message Screen. The probable failing board can be determined from the type of error indicated on that screen. The possible error types and their corresponding probable areas of failure are listed in Table 4.1.

NOTE

> While the boards listed in Table 4.1 are the ones most likely to be faulty in a given instance, it is always possible that another board or boards are involved.

If an abnormal processor halt occurs during diagnostic execution, preventing the display of error messages, the octal value displayed in the control panel lights can also be used to isolate the probable area of failure. This value corresponds to the number of the individual subtest running when the processor halted. Correlation of this number with the test set that was running when the processor halted (displayed in inverse video on the Diagnostic Control Screen) yields the probable failing board. These numbers are indexed to their probable error types and areas of failure in Table 4.2.

In test sequences encountering multiple errors, the totals displayed by the Error Log Screen may also suggest, in a rough way, the probable areas of failure. However, it is always advisable to solve errors in the order in which they appear on the Error Message Screen, because the test sets are built on the assumption that the previous tests have passed.

Even when the processor under test is not generating immediate errors, it is advisable to let General Test loop through a number of passes in order to detect intermittent or isolated errors. If such errors are occurring, a single pass might not encounter them.

PART 6A - GENERAL TEST
ERROR CONDITIONS AND DISPOSITIONS

If, during General Test, a flashing error message appears immediately beneath the banner on any of the three screens, this indicates that a critical error has occurred, not of the kind detected by General Test, but in the functioning of the OSP itself. Invalid data may have been entered, or an error may have occurred in the Diagnostic Data Transceiver (DDT) communication path. A return to the startup and self-test procedures may be indicated

Table 4.1 Error Types and Probable Areas of Failure

ERROR TYPE	PROBABLE AREA OF FAILURE
ILLEGAL VECTOR INTO ERROR TABLE	IPU
INTERNAL FUNCTIONS	IPU
INTERNAL REGISTERS	IPU
INTERNAL INTERRUPTS	IPU
PROCESSOR STATUS REGISTER	IPU
J/K BUS PARITY DETECTION	IPU
PARITY VALID CIRCUITRY	IPU
ROM DATA PARITY DETECTION	IPU, CCD
ROM ADDRESS PARITY DETECTION	IPU ,CCD
EPT CHECKSUM ERROR: ADDRESS XXXXXX, BAD WORD XXXXX	IPU
SEPT VALIDATION ERROR	IPU
IEPT VALIDATION ERROR	IPU
MAP ARRAY	MCB
MAPA PARITY DETECTION	MCB
PMA PARITY DETECTION	MCB
PMD PARITY DETECTION	MCB
MAP DATA PARITY DETECTION	MCB
MDIAG PARITY DETECTION	MCB
MEMORY DATA PATH	MCB
MEMORY ACCESS	MCB
MEMORY ABSENCE	MCB
MEMORY ADDRESSING	MCB

Table 4.1 Error Types and Probable Areas of Failure (Cont'd)

ERROR TYPE	PROBABLE AREA OF FAILURE
MAP REWRITE	MCB
ADDRESS BREAKPOINT	MCB
MEMORY ERROR CORRECTION	MCB
UCME/CME HANDLING	MCB
INTERVAL CLOCK	MCB
OUTQ SELECTION/LOADING	MCB
INQ SELECTION/INTERRUPT	MCB, XBUS, YBUS
XBUS PATTERN TEST	MCB, XBUS
YBUS PATTERN TEST	MCB, YBUS
XBUS EXTENDED PATTERN	MCB, XBUS
YBUS EXTENDED PATTERN	MCB, YBUS
NONCONTIGUOUS MEMORY FOUND FROM PAGE XXXXX TO PAGE XXXXX	MCB, Memory*
INTERRUPT VECTORING	IPU, MCB
INSTRUCTION END VECTORING	IPU, MCB, Memory*
CHANNEL INTERNAL FUNCTIONS	CCD
CHANNEL CBUS PARITY DETECTION	CCD
CHANNEL CCSD PARITY DETECTION	CCD
CHANNEL I/O BUS COMMUNICATION	CCD, I/O BUS
CHANNEL MEMORY ACCESS	CCD, MCB, Memory*
CONTROL STORE ACCESS ERROR	CCD
LCS ERROR: ADDRESS XXXXX, EXPECTED XXXXXXXXXXXX ACTUAL XXXXXXXXXXXX	CCD

*If this error occurs, it is advisable to run the Memory Diagnostic to determine whether the problem is memory-related.

Table 4.2 Interpretation of Control Panel Lights

VALUE IN CONTROL PANEL LIGHTS (octal)	ASSOCIATED ERROR TYPE AND PROBABLE AREA OF FAILURE
0	Internal Functions (IPU) Map Array (MCB) Interrupt Vectoring (MCB, IPU)
1	Internal Functions (IPU) Map Array (MCB) Instruction End Vectoring (IPU, MCB, Memory)
2	Internal Registers (IPU) MAPA Parity Detection (MCB) Channel CBUS Parity Detection (CCD)
3	Internal Registers (IPU) PMA Parity Detection (MCB) Interrupt Vectoring (IPU, MCB)
4	Internal Registers (IPU) PMD Parity Detection (MCB) Interrupt Vectoring (IPU, MCB) Channel Internal Functions (CCD)
5	Internal Functions (IPU) Parity Valid Circuitry (MCB) Interrupt Vectoring (IPU, MCB) Channel Internal Functions (CCD)
6	Internal Registers (IPU) Parity Valid Circuitry (MCB) Interrupt Vectoring (IPU, MCB) Channel Internal Functions (CCD)
7	Internal Registers (IPU) Map Data Parity Detection (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
10	Internal Registers (IPU) MDIAG Parity Detection (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
11	Internal Registers (IPU) Memory Data Path (MCB) Channel I/O Bus Communication (CCD, I/O Bus)

Table 4.2 Interpretation of Control Panel Lights (Cont´d)

VALUE IN CONTROL PANEL LIGHTS (octal)	ASSOCIATED ERROR TYPE AND PROBABLE AREA OF FAILURE
12	Internal Functions (IPU) Memory Access (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
13	Internal Functions (IPU) Memory Absence (MCB) Channel Internal Functions (CCD)
14	Internal Functions (IPU) Memory Addressing (MCB) Channel Internal Functions (CCD)
15	Internal Functions (IPU) Map Rewrite (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
16	Internal Functions (IPU) Channel Internal Functions (CCD)
17	Internal Functions (IPU) Channel Internal Functions (CCD)
20	Internal Functions (IPU) Map Rewrite (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
21	Internal Functions (IPU) Map Rewrite (MCB) Channel I/O Bus Communication (CCD, I/O Bus)
22	Internal Interrupts (IPU) Map Rewrite (MCB) Channel Memory Access (CCD,MCB, Memory)
23	Internal Functions (IPU) Map Rewrite (MCB) Channel Memory Access (CCD, MCB, Memory)
24	Processor Status Register (IPU) Map Rewrite (MCB) Channel Memory Access (CCD, MCB, Memory)

Table 4.2 Interpretation of Control Panel Lights (Cont´d)

VALUE IN CONTROL PANEL LIGHTS (octal)	ASSOCIATED ERROR TYPE AND PROBABLE AREA OF FAILURE
25	J/K Bus Parity Detection (IPU) Map Rewrite (MCB) Channel Memory Access (CCD, MCB, Memory)
26	Parity Valid Circuitry (IPU) Map Rewrite (MCB) Channel Memory Access (CCD, MCB, Memory)
27	ROM Data Parity Detection (IPU, CCD) Address Breakpoint (MCB) Channel Memory Access (CCD, MCB, Memory)
30	ROM Address Parity Detection (IPU, CCD) Channel Memory Access (CCD, MCB, Memory)
31	EPT Checksum Error (IPU) Channel Memory Access (CCD, MCB, Memory)
32	SEPT Validation Error (IPU) Channel Memory Access (CCD, MCB, Memory)
33	IEPT Validation Error (IPU) Channel CBUS Parity Detection (CCD)
34	Channel CCSD Parity Detection (CCD)
35	ROM Address Parity Detection (IPU, CCD)
36	ROM Data Parity Detection (IPU, CCD) Control Store Access Error (CCD)
37	ROM Address Parity Detection (IPU, CCD) Control Store Access Error (CCD)
40	J/K Bus Parity Detection (IPU) Memory Error Correction (MCB) LCS Error (CCD)
41	J/K Bus Parity Detection (IPU) Memory Error Correction (MCB) LCS Error (CCD)

Table 4.2 Interpretation of Control Panel Lights (Cont´d)

VALUE IN CONTROL PANEL LIGHTS (octal)	ASSOCIATED ERROR TYPE AND PROBABLE AREA OF FAILURE
42	Memory Error Correction (MCB)
43	UCME/CME Handling (MCB)
44	UCME/CME Handling (MCB)
45	Internal Functions (MCB)
46	Internal Functions (MCB)
47	Interval Clock (MCB)
50	Interval Clock (MCB)
51	MAPA Parity Detection (MCB)
52	PMA Parity Detection (MCB)
53	PMD Parity Detection (MCB)
54	MDIAG Parity Detection (MCB)
55	Map Data Parity Detection (MCB)
61	OUTQ Selection/Loading (MCB)
62	OUTQ Selection/Loading (MCB)
63	INQ Selection/Interrupt (MCB, XBUS, YBUS)
64	XBUS Pattern Test (MCB, XBUS)
65	YBUS Pattern Test (MCB, YBUS)
66, 70	XBUS Extended Pattern (MCB, XBUS)
67, 71	YBUS Extended Pattern (MCB, YBUS)
72	Noncontiguous Memory Found (MCB, Memory)

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES

To run General Test, proceed according to the following sequence of steps:

a. Depress shifted function key F6 to select the Processor Status Screen on the OSP. (Refer to Figure 5-1.)

   1. Enter at the top of the screen the number of the processor to be tested.

   2. Depress function key F1.

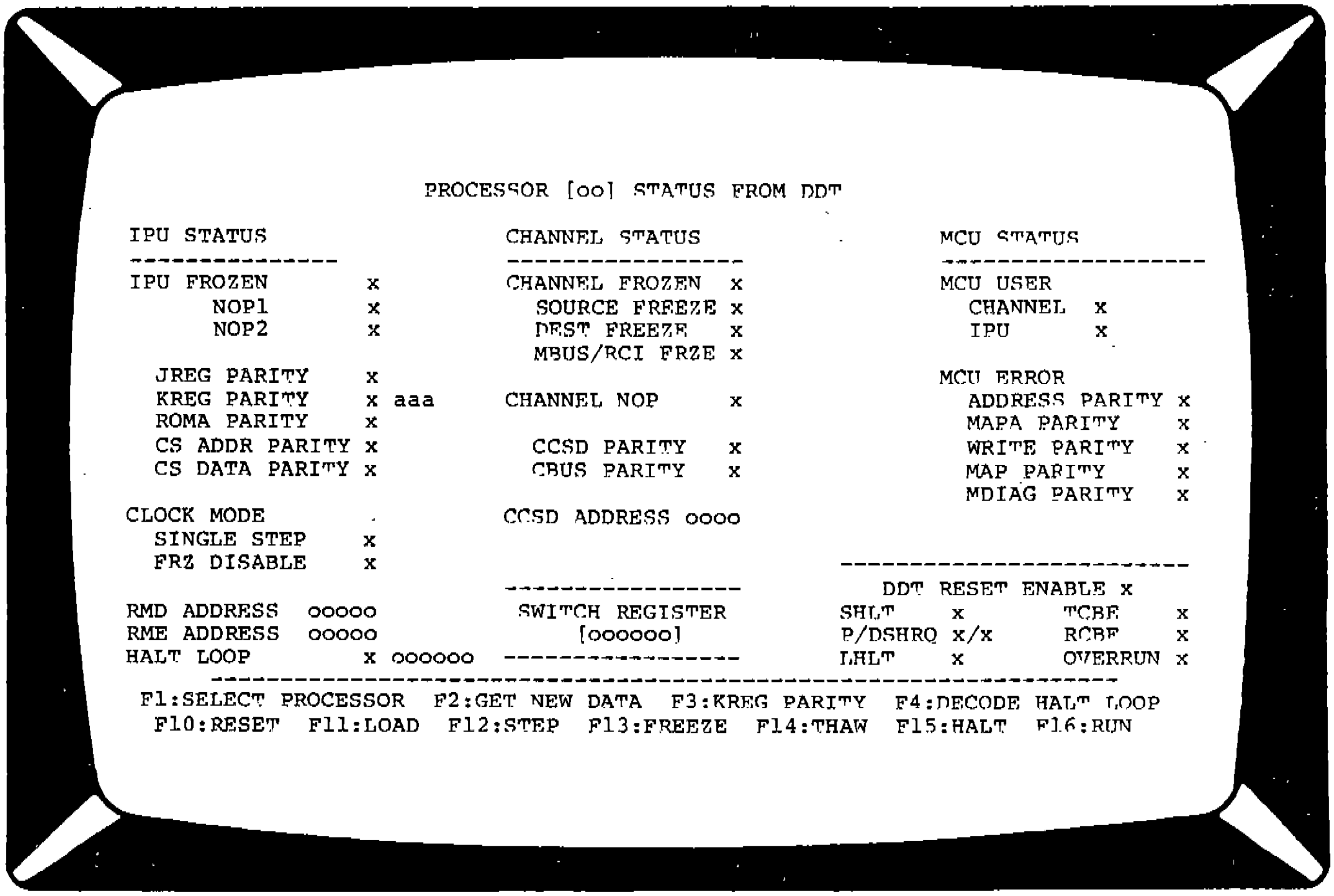
b. Depress shifted function key F7 to fetch the Microdiagnostic Loader Screen on the OSP. (Refer to Figure 5-2.)

   1. Locate General Test in the list of diagnostic file names on the Microdiagnostic Loader Screen.

   2. Depress the function key whose number corresponds to the number assigned General Test in the list of file names. (The message handler for General Test is now downloading, and the OSP switches automatically to the Diagnostic Control Screen.)

NOTE

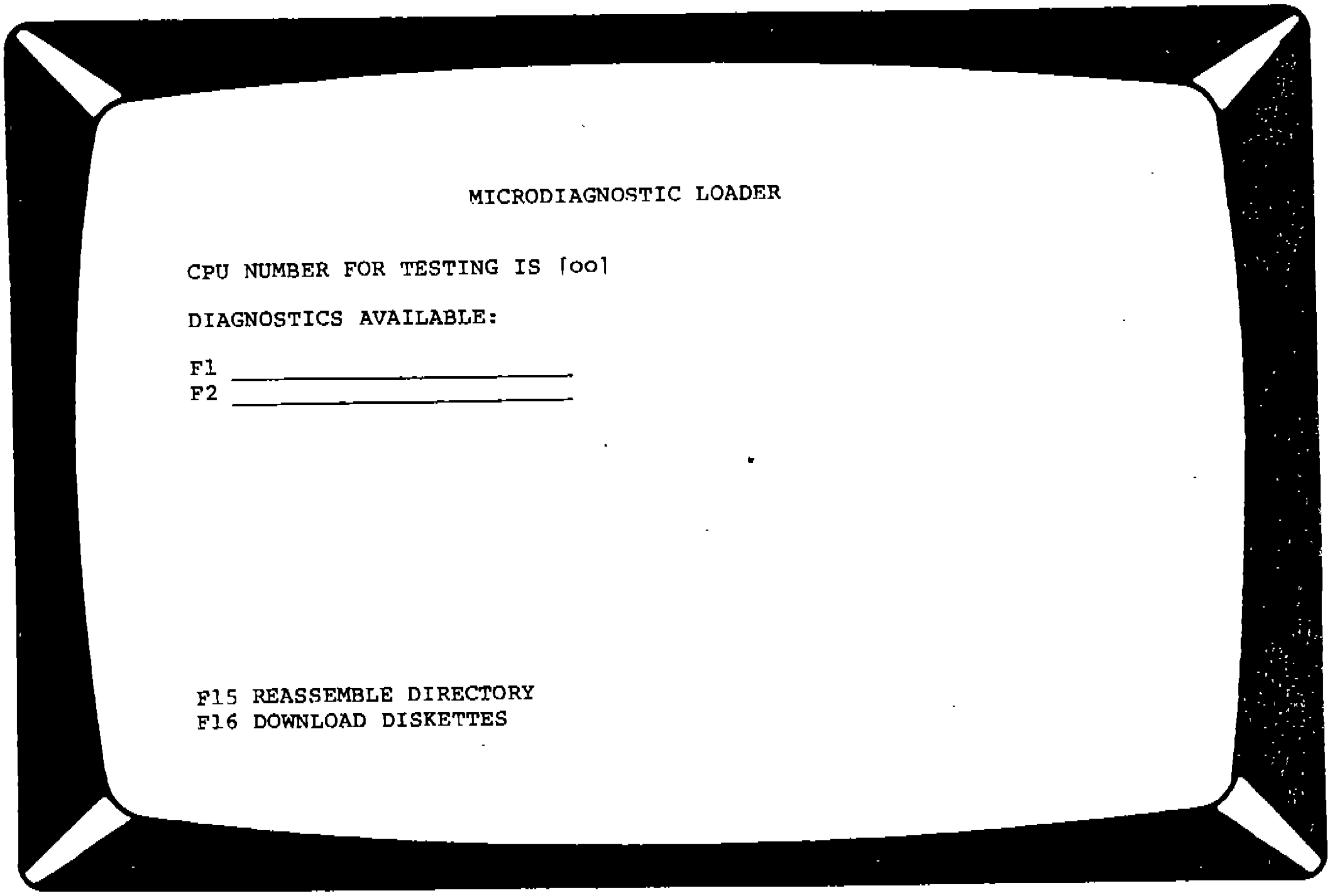
For the diagnostic to be downloaded, the processor under test must be in a halt loop. The message handler accomplishes this automatically by performing a reset of the processor before loading the microdiagnostic set. However, for the message handler to do this, the RESET ENABLE switch must be set on the Processor Maintenance Interface (PMI) panel. If this switch is disabled when the diagnostic load is attempted, an error message will appear on the screen. Refer to paragraph 3.3.3 of the OSP User Guide, 82801-B00 , for details of resetting a processor.



```
                     PROCESSOR [oo] STATUS FROM DDT

IPU STATUS                CHANNEL STATUS                  MCU STATUS
---------------           -----------------               -------------------
IPU FROZEN       x        CHANNEL FROZEN  x               MCU USER
      NOP1       x          SOURCE FREEZE x                 CHANNEL   x
      NOP2       x          DEST FREEZE   x                 IPU       x
                            MBUS/RCI FRZE x
  JREG PARITY    x                                        MCU ERROR
  KREG PARITY    x aaa    CHANNEL NOP     x                 ADDRESS PARITY x
  ROMA PARITY    x                                          MAPA PARITY    x
  CS ADDR PARITY x          CCSD PARITY   x                 WRITE PARITY   x
  CS DATA PARITY x          CBUS PARITY   x                 MAP PARITY     x
                                                            MDIAG PARITY   x
CLOCK MODE                CCSD ADDRESS oooo
  SINGLE STEP    x
  FRZ DISABLE    x                                  -------------------------
                          -----------------            DDT RESET ENABLE x
RMD ADDRESS  ooooo          SWITCH REGISTER         SHLT      x       TCBE    x
RME ADDRESS  ooooo             [oooooo]             P/DSHRQ  x/x      RCBF    x
HALT LOOP        x oooooo -----------------         LHLT      x       OVERRUN x
      -------------------------------------------------------------------
 F1:SELECT PROCESSOR  F2:GET NEW DATA  F3:KREG PARITY  F4:DECODE HALT LOOP
  F10:RESET  F11:LOAD  F12:STEP  F13:FREEZE  F14:THAW  F15:HALT  F16:RUN
```

T16/8116A-04

Figure 5-1 Processor Status Screen



```
                    MICRODIAGNOSTIC LOADER

CPU NUMBER FOR TESTING IS [00]

DIAGNOSTICS AVAILABLE:

F1 ________________________
F2 ________________________






 F15 REASSEMBLE DIRECTORY
 F16 DOWNLOAD DISKETTES
```

T16/8116A-05

Figure 5-2 Microdiagnostic Loader Screen

c. Scan the run-time parameters on the Diagnostic Control Screen.

d. If the settings of the run-time parameters, Halt on Error, Loop on Test, and Diagnostic Test Number are not appropriate, change them as required.

To change a run-time parameter

1. Depress carriage return until the cursor advances to the control field to be changed.

2. Enter the desired change. (The changes are honored by the message handler only when the test starts.)

e. When the modifiable fields are set as desired, depress function key F9 to start the test running.

NOTE

The selected test completes at least one pass without further necessary action from the operator, unless Halt on Error is set to Y and an error is detected. In that case, the test suspends execution, DIAGNOSTIC STATE on all screens changes to DIAGNOSTIC SUSPENDED, and the message handler awaits instructions. Run time for a full pass of all tests is about five minutes. When a pass of the selected test is completed, the test automatically updates Processor Configuration and PASSCOUNT on the Diagnostic Control Screen. If an error is encountered during the pass, ERROR DETECTED flashes on the Diagnostic Control Screen, and pertinent information is displayed in the automatic updates of the Error Message and Error Log screens. Unless Halt on Error is set to Y and Loop on Test to N, the test continues to loop and updates with appropriate information at the end of each pass.

f. When an error or errors have been detected, perform the following:

1. Depress function key F2 to fetch the Error Message Screen. (Refer to paragraph 3.1.2 for a description of this screen.)

2. Depress F9 to update the screen. (The probable area of failure can be determined from the type or types of errors indicated on this screen.)

3. Locate the detected error or errors and their associated probable areas of failure in Table 4.1.

NOTE

It is often possible at this point to deduce which board is failing, but it is advisable to correlate this deduction with the information available on the Error Log Screen, particularly if the Error Message Screen is full (seven errors). While the Error Message (or Error Log) Screen is being examined, the executing test may or may not be suspended, as desired. If not suspended, it continues to loop, updating at the end of each pass. To suspend the test, depress F10 from the Diagnostic Control Screen.

g. Depress function key F3 to fetch the Error Log Screen. (The Error Log Screen displays the total number of errors detected in each of the four test sets.)

h. Depress F9 to update the Error Log Screen.

NOTE

Each of the four test sets is designed to test functions associated with a particular board. The Instruction Processing Tests test the IPU. The Memory Control and Interprocessor Bus Tests test the MCB. The Control Store and I/O Channel Tests test the CCD. The Instruction Cycling Tests test both IPU and MCB functions. Therefore, in tests that encounter numerous errors, it is sometimes easier to deduce from the Error Log totals which board or boards are failing. It is always important to know what test set is running when a given error is encountered.

i. Analyze the information from both the Error Message and the Error Log Screens to decide which board is the most likely to be faulty.

j. Replace the suspected faulty board.

CAUTION

Before replacing a board, power down the processor under test. For details of the board replacement procedure, refer to section 5 of the Tandem Processor Subsystems Maintenance Manual, T16/8809.

k. After replacing the board under suspicion, rerun General Test to ensure that the failure is eliminated. The rerun procedure is as follows:

  1. Power the processor under test back on. (Refer to section 5 of the Processor Subsystems Maintenance Manual, 82809-A01, for details.)

  2. Depress F16 (to reinitialize General Test).

  3. Reset run-time parameters as desired.

  4. Depress F9 (to start the diagnostic running).

l. If the same errors persist, or if a different error occurs, another faulty board may be indicated, or two boards may have been faulty initially.

  1. Return to the information on the Error Message and Error Log Screens to decide the second most probable failing board.

  2. Repeat the replacement procedure with the second board, again rerunning General Test to see whether the failures are eliminated.

NOTE

> As indicated in Table 4.1, an error type can suggest more than one probable area of failure, particularly errors associated with the Instruction Cycling Tests. Note too that an error like CHANNEL I/O BUS COMMUNICATION can suggest not only a faulty CCD board, but also a faulty I/O controller.

m. If errors persist after two boards have been replaced, report the problem immediately to Tandem regional headquarters.

5.1 SPECIAL CONSIDERATIONS

It is possible to run General Test in a number of processors at the same time. However, each test must be started separately. To do this, observe the following procedure:

a. Start General Test in the first processor to be tested, as outlined in steps a through e of paragraph 5.0.

b. With General Test started in this processor, return to the PROCESSOR STATUS screen by depressing shifted function key F6. Select the next processor to be tested, and repeat steps a through e of paragraph 5.0.

This procedure may be repeated for as many as four processors. Once started, all will continue running until terminated by manual intervention. While several General Tests may, in this way, be running simultaneously, only one processor can be actively monitored on the General Test screens at any given time. The one being monitored is governed by the processor number currently selected on the Processor Status Screen. For further information, refer to paragraph 3.3.5 of the OSP User Guide, 82801-B00.

It is not possible, on the other hand, to run more than one type of downloaded diagnostic at the same time. Due to space limitations, OSP RAM can accommodate the microcode for only one message handler at a time. Thus, while the message handler for General Test, once downloaded, can direct the running of General Test in as many as four processors at once, if a different diagnostic (such as the Memory Diagnostic) is to be run, General Test must first be stopped in all the processors where it is running. Only then can the message handler for the new diagnostic be downloaded. To prevent an attempted downloading of a second diagnostic while another diagnostic is already running in one or more processors, the Microdiagnostic Loader Screen is unavailable as long as a diagnostic test is running in any processor. The procedure for terminating multiple tests running simultaneously is as follows:

a. From the Diagnostic Control Screen for the currently monitored processor, perform the following:

   1. Depress function key F10 to suspend the diagnostic.

   2. Depress function key F8 to terminate the diagnostic.

NOTE

> An executing test must be suspended in order to clear the error lights on the MCB board; otherwise they are in a random state. If the diagnostic is running in more than one processor, depressing F8 not only terminates the test in the currently selected processor, it also calls up the Diagnostic Control Screen of the next processor in which the diagnostic is running.

b. When the Diagnostic Control Screen of the next processor in which the test is running appears, first depress F10 to suspend the diagnostic, and then F8 to terminate it.

c. Repeat step b until all currently executing diagnostics are terminated.

NOTE

> When the last executing diagnostic is terminated, the final depression of F8 calls up the Processor Status Screen. From this screen, a new processor may then be selected for test, as outlined in paragraph 5.0, step a. The procedure for running a different diagnostic in the newly selected processor is the same as that outlined in paragraph 5.0, step b.

PART 6B

Memory Diagnostic

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 1, PART 6B

MEMORY DIAGNOSTIC

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95014

Tandem Part Number 82803

December 1982
Printed in U.S.A.

REVISION RECORD

REVISION	DATE	DESCRIPTION
A	Aug. 1981	First Edition
B	Dec. 1982	Editorial changes.

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 1, CHAPTER 1, PART 6B

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION..1-1

SECTION 2
OVERVIEW

2.0 OVERVIEW..2-1
2.1 PROGRAM INITIALIZATION..2-1
2.2 PROGRAM EXECUTION..2-2
2.3 EXAMINATION OF ERROR MESSAGES..2-3
2.4 REINITIALIZATION OF THE DIAGNOSTIC..2-3
2.5 DIAGNOSTIC TEST LIST..2-3
2.5.1 Memory Addressing Test..2-3
2.5.2 Marching Pattern Test..2-4
2.5.3 Moving Inversions Test..2-4
2.5.4 Sliding Diagonals Test..2-4
2.5.5 Power Fail Test..2-5
2.5.6 Refresh Delay Test..2-5

SECTION 3
COMMANDS

3.0 COMMANDS..3-1
3.1 SCREEN SELECTION AND MANIPULATION..3-1
3.1.1 Diagnostic Control Screen..3-2
3.1.2 Error Message Screen..3-4
3.1.3 Error Log Screen..3-6
3.2 FUNCTION KEY COMMANDS USED IN MEMORY DIAGNOSTIC..........3-9

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS..4-1

TABLE OF CONTENTS

Par No. Title Page No.

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES..5-1
5.1 SPECIAL CONSIDERATIONS..................................5-10

LIST OF ILLUSTRATIONS

Fig.No. Title Page No.

3-1 Diagnostic Control Screen........................3-3
3-2 Error Message Screen.............................3-5
3-3 Board Log Table..................................3-7
3-4 Bit Log Table....................................3-8

5-1 Processor Status Screen..........................5-2
5-2 Microdiagnostic Loader Screen....................5-3

LIST OF TABLES

Table No. Title Page No.

3.1 Screens Available in Memory Diagnostic...........3-1
3.2 Function Key Commands Used in Memory Diagnostic.................................. 3-9

4.1 Types of Errors on the Message Screen............4-1

PART 6B - MEMORY DIAGNOSTIC
TABLE OF CONTENTS

PART 6B - MEMORY DIAGNOSTIC
TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The Tandem NonStop II (TM) Memory Diagnostic (T9466A00.MEM2420) tests the main memory of the system to ensure that it is functioning properly. When a Tandem memory board or boards are generating errors, the Memory Diagnostic can be run to determine the error type and to isolate the failure to the chip level. After the faulty chip or chips have been replaced, the diagnostic can be run again as an endtest to ensure that the errors have been eliminated.

The Memory Diagnostic is a downloadable program. It resides on one of the two flexible (floppy) discs associated with the Operations and Service Processor (OSP) and is run through the OSP. The Memory Diagnostic consists of six different tests:

a. Memory addressing

b. Marching pattern

c. Moving inversions

d. Sliding diagonals

e. Power fail

f. Refresh delay

These tests can be run in any sequence or combination. They can be used to determine not only whether memory is malfunctioning but, if so, precisely which chip or chips are failing. The approximate run time per 512 kilobytes of memory is as follows: memory addressing - 2 sec; marching pattern - 45 sec; moving inversions - 12 min; sliding diagonals - 2.9 hr; power fail - 1 sec (plus 1 min for reloading the diagnostic); refresh delay - 20 sec. These times may vary considerably, depending on such factors as number and kinds of errors actually detected, OSP workload, etc.

The Memory Diagnostic is designed to run on all NonStop II Operating Systems beginning with revision A00.

PART 6B - MEMORY DIAGNOSTIC
INTRODUCTION

Operator interface to the Memory Diagnostic occurs through one of three screens generated by the special message handler for the diagnostic. They are displayed at the OSP terminal and manipulated by the use of predefined function keys. These screens are:

a. Diagnostic Control Screen

b. Error Message Screen

c. Error Log Screen

Two error tables are available on the Error Log Screen: the Board Log Table and the Bit Log Table.

For a complete description of the OSP, its functions, and its specific relation to the downloadable diagnostics, refer to Tandem OSP User Guide, 82801.

SECTION 2
OVERVIEW

2.0 OVERVIEW

The OSP must be functioning properly in order to run the Memory Diagnostic.

Before running the Memory Diagnostic, the processor to be tested must first be selected on the Processor Status Screen of the OSP. The Memory Diagnostic is then accessed through the Microdiagnostic Loader Screen of the OSP.

When the Memory Diagnostic is selected from the diagnostics available on this screen, the code for its special message handler is immediately loaded from floppy disc into OSP RAM. The message handler enters an idle loop and prevents the execution of any tests until function key F16 is depressed to load the memory diagnostic itself. When F16 is depressed, the message handler reads the appropriate diagnostic microcode from floppy disc and loads it into Loadable Control Store (LCS) of the processor under test.

2.1 PROGRAM INITIALIZATION

Once the diagnostic has been loaded into LCS, the message handler performs a startup of the diagnostic by forcing the LCS sequencer to jump to the initialization routine. The diagnostic is then loosely coupled to the message handler and requires only occasional monitoring by it. This monitoring or query function consists of a diagnostic state request message which is periodically sent to the diagnostic by the message handler. The diagnostic responds with one of three hex ASCII digits (0, 1, or 2). These digits represent one of the three possible states of program execution (diagnostic executing, execution complete, or execution suspended, respectively).

After control is passed to the initialization routine, it performs a setup of the operating environment, determines the size of physical memory, and reports the upper and lower limits found to exist. It also checks for noncontiguous memory and, if any is found, returns an error message indicating the section of memory that is noncontiguous. The check for noncontiguous memory proceeds until all addressable memory space has been scanned. When the initialization procedure is complete, control passes to a command interpreter which establishes the link between the diagnostic in the processor and the message handler in the OSP. The message handler enters an idle loop and awaits operator instructions.

If function key F16 is depressed after the diagnostic is loaded into LCS, the message handler first verifies whether the diagnostic microcode is present. If the diagnostic is already present, depressing function key F16 has the effect of simply starting the initializing routine. If the diagnostic is lost, the microcode is first reloaded and then initialization is started.

2.2 PROGRAM EXECUTION

At startup, the message handler defaults to the Diagnostic Control Screen. This screen automatically displays current information about the size of memory, the status of diagnostic execution, and the detection of errors in the memory under test. It also enables alteration of the memory address range to be tested, modification of certain run-time parameters, and selection of the particular memory test or test sequences to be run.

When the fields have been modified as desired and the START TEST function key (F9) has been depressed, the message handler reads the supplied information and sets up and runs the test accordingly.

If an error is detected during execution of a test an error handler routine in the diagnostic responds by sending an error message to the message handler. Errors are logged by address and bit position in a table maintained in LCS. If the List Errors control field is enabled, error information is displayed on the Error Message Screen as well as on the Error Log Screen.

The status of the memory test after transmission of an error message depends on the setting of the Halt on Error control field. If Halt on Error is disabled, execution of the test resumes at the point of error detection. If Halt on Error is enabled, the error handler passes control to the command interpreter and the diagnostic is suspended.

2.3 EXAMINATION OF ERROR MESSAGES

While the selected memory test is executing, any of the other screens can be examined for information about errors being encountered. Before the Error Message Screen, the Board Log Table, or the Bit Log Table are examined, the executing memory test may or may not be suspended, as desired. This can be done from the Diagnostic Control Screen. If the executing test is not suspended, it continues to loop and updates the relevant screens with appropriate information.

2.4 REINITIALIZATION OF THE DIAGNOSTIC

The Memory Diagnostic can be reinitialized at any time from any of the three screens, by depressing the appropriate function key. Reinitialization terminates the diagnostic being run, clears the Error Message Screen and Error Log Tables of accumulated information, redisplays the Diagnostic Control Screen if in another screen, returns the run-time parameters and passcount to their default settings, redetermines the size of memory, and places the message handler in the idle loop where it waits for operator instructions.

2.5 DIAGNOSTIC TEST LIST

The operator test selection list, which appears on the Diagnostic Control Screen, is organized into individual tests and test sequences. The test sequences are designed to meet two requirements; namely, to isolate defective memory chips, and to quickly qualify replacement chips. This test philosophy stems from the time required to perform comprehensive tests (e.g., the Sliding Diagonals Test requires 2.9 hr. per 512 kb of memory) and from the need for a quick functional test of a repaired memory board. The individual tests are described in detail in paragraphs 2.5.1 through 2.5.6.

2.5.1 Memory Addressing Test

The Memory Addressing Test executes two passes. The first pass writes the lower 16 bits of the address under test into itself. After all locations within the addressing range have been loaded, the addresses are read back and verified. The second pass writes the physical page number of the address into itself, writes all locations, then reads back and verifies.
Execution time - approx 2 sec per 512 kb.

2.5.2 Marching Pattern Test

The Marching Pattern Test writes a test pattern in all locations within the addressing range and reads each location to verify that it contains the pattern. Each location is then written with the complement of the pattern and read to verify that it contains the complement. Finally, each location is rewritten with the original pattern. This test is performed in ascending and descending addresses within the address range. The initial test pattern used is all zeros. On successive passes, the pattern is circularly shifted left, with the end wrap-around bit inverted.
Execution time - approx 45 sec per 512 kb.

2.5.3 Moving Inversions Test

The Moving Inversions Test writes a test pattern in all locations within the addressing range. Each test address is first read and verified to contain the previous (old) pattern, written with the current (new) pattern, and then read and verified for the new pattern. The test address sequence is obtained by incrementing in a given address bit position and adjusting so each location is tested exactly once per data pattern. A similar reverse sequence decrements in the given bit position. Finally, the address inc/dec bit position cycles so that each address bit is exercised in both directions for all combinations of the other address bits. The initial test pattern used is all zeros. On successive passes, the pattern is circularly shifted left, with the end wrap-around bit inverted.
Execution time - approx 12 min per 512 kb.

2.5.4 Sliding Diagonals Test

The Sliding Diagonals Test writes a test pattern in all locations within the addressing range, substituting the complement of the pattern at every fifth location. The pattern written in memory is then read and verified. Next, the whole pattern in memory is shifted up one address location and compared in a series of write/read passes until all five pattern positions have been verified. Then a new pattern is written into memory, with the complement pattern in every ninth location. This pattern is also shifted in memory and checked for in all nine pattern positions. The process is repeated similarly for spacings of 17, 33, 65, and 129. The initial test pattern used is all zeros. On successive passes, the pattern is circularly shifted left, with the end wrap-around bit inverted.
Execution time - approx 2.9 hrs per 512 kb.

2.5.5 Power Fail Test

The Power Fail Test writes a test pattern in all locations within the addressing range. The operator is then instructed to power the processor under test down and up, leaving battery backup on. When the processor power is restored , the test then verifies that the contents of memory are unchanged.
Execution time - approx 1 sec per 512 kb. (plus 1 min for reloading the diagnostic)

2.5.6 Refresh Delay Test

The Refresh Delay Test writes a test pattern in all locations within the addressing range. Refresh is then disabled for the interval specified by the refresh delay parameter within the diagnostic control field. The test then re-enables the refresh counter and verifies that the contents of memory are unchanged. The initial test pattern used is all zeros. On successive passes, the pattern is circularly shifted left, with the end wrap-around bit inverted.
Execution time - approx 20 sec per 512 kb.

PART 6B - MEMORY DIAGNOSTIC OVERVIEW

SECTION 3
COMMANDS

3.0 COMMANDS

The NonStop II Memory Diagnostic is accessed and controlled through preformatted screens and predefined function keys on the OSP. Operator interface does not take place directly through a command interpreter but through a message handler that generates the screens and defines the various function keys needed to manipulate them.

3.1 SCREEN SELECTION AND MANIPULATION

Once the Memory Diagnostic is selected from the Microdiagnostic Loader Screen and commanded to run, the message handler defaults to the Diagnostic Control Screen. After this point, any one of the three available screens can be fetched from any other, by depressing the function key associated with the desired screen. From each of the given screens, the necessary test operations can then be performed by other function keys defined in the message handler. A special function key (F8) terminates the Memory Diagnostic in the current processor and returns either to the Memory Diagnostic running in a different processor or to the Processor Status Screen. For a description of the use of this function key command when the Memory Diagnostic is running in more than one processor at the same time, refer to paragraph 5.1. For further information, refer to paragraph 3.3.5 of the OSP User Guide, 82801.

The screens and their associated function keys are listed in Table 3.1. The formats of the Memory Diagnostic screens and the specific function key commands executable from each are described in paragraphs 3.1.1 through 3.1.3.

Table 3.1 Screens Available in Memory Diagnostic

Function Key	Screen Displayed
F1	Diagnostic Control Screen
F2	Error Message Screen
F3	Error Log Screen

3.1.1 Diagnostic Control Screen

The Diagnostic Control Screen is invoked by depressing function key F1. The format of the Diagnostic Control Screen is shown in Figure 3-1.

At the top of the Diagnostic Control Screen is a banner identifying the processor number, diagnostic name, and release date. At the bottom is a summary of the relevant function keys and their uses. The remainder of the screen is divided into three active fields: Area Under Test, Diagnostic Control, and DIAGNOSTIC TEST NUMBER LIST.

During diagnostic initialization, Area Under Test is automatically set to reflect the upper and lower limits of memory found to exist by the initialization routine. These limits can be modified before starting a test.

The Diagnostic Control field contains five parameters which can likewise be altered according to the needs of the situation. The default settings are: Halt on Error and List Errors set to N (disabled); Loop on Test and Access Margin set to Y (enabled); and Refresh Delay set to 10 ms.

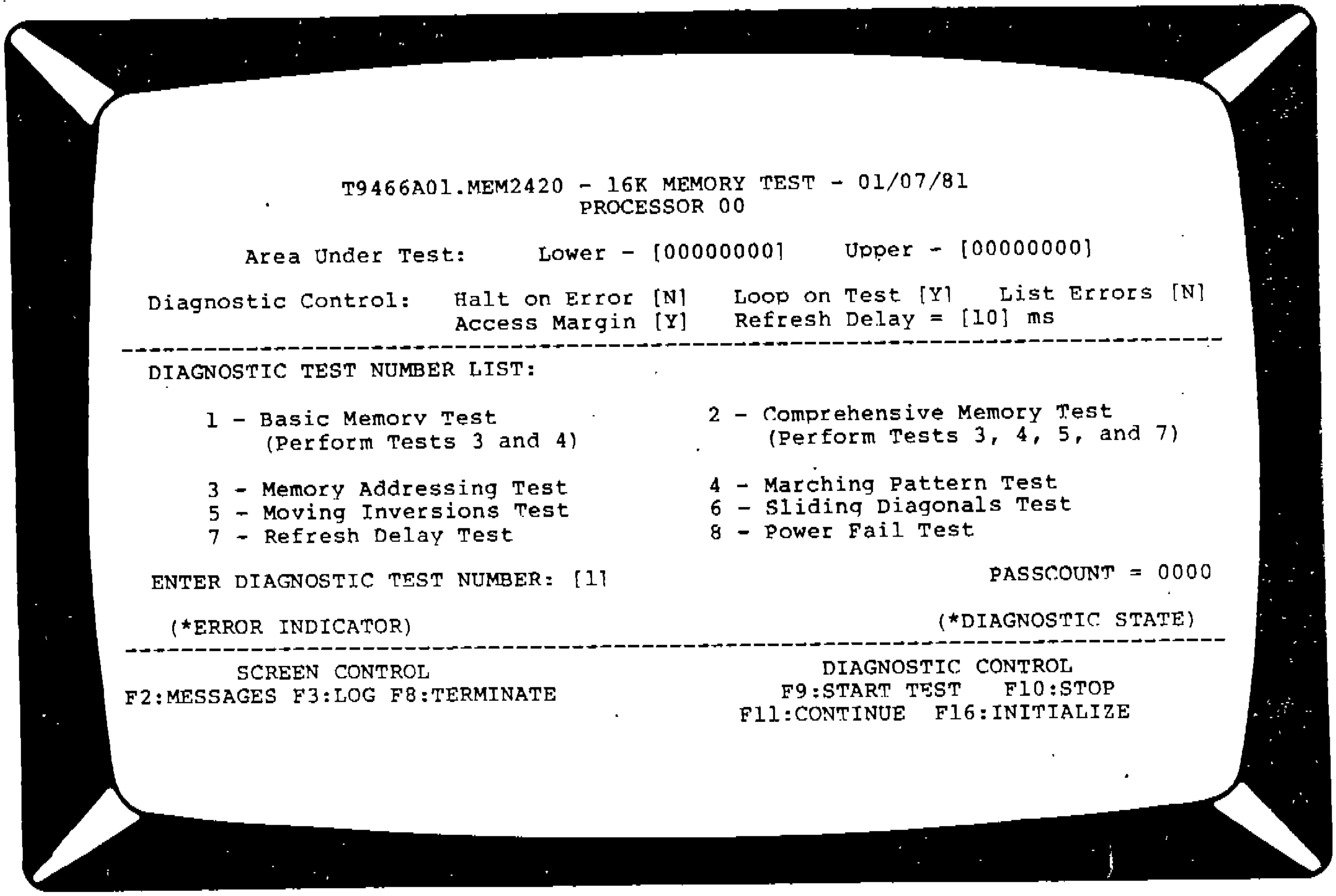
If Halt on Error is enabled, the diagnostic suspends execution on encountering an error. If List Errors is disabled, the message handler does not receive error messages when the diagnostic encounters an error and, consequently, does not display error information on the Error Message Screen. If Loop on Test is enabled, the selected diagnostic test restarts automatically on completion. If Access Margin is enabled, the memory access time data window is shortened to assist in the discovery of marginal chips. The setting of Refresh Delay determines the interval at which the Refresh Delay Test runs.

The settings of Halt on Error and List Errors are not entirely independent of each other. If Halt on Error is enabled, List Errors is automatically enabled also, regardless of the setting as displayed or as changed by the operator.

DIAGNOSTIC TEST NUMBER LIST identifies the available tests or test sequences and assigns a number to each. The number of the test to be run must be entered after ENTER DIAGNOSTIC TEST NUMBER. This field defaults to the Basic Memory Test if no test number is entered.

The DIAGNOSTIC TEST NUMBER LIST also provides ERROR INDICATOR, DIAGNOSTIC STATE, and PASSCOUNT fields.

The ERROR INDICATOR and DIAGNOSTIC STATE fields are automatically updated by the message handler as the diagnostic runs. When the Memory Diagnostic is first invoked from the Microdiagnostic Loader Screen, ERROR INDICATOR displays NO ERRORS DETECTED and DIAGNOSTIC STATE flashes DIAGNOSTIC LOADING. As soon as the diagnostic is loaded and control passes to the initialization routine, DIAGNOSTIC STATE changes to display DIAGNOSTIC INIT. When initialization is complete, DIAGNOSTIC STATE again changes, to DIAGNOSTIC IDLE, which indicates



```
              T9466A01.MEM2420 - 16K MEMORY TEST - 01/07/81
                              PROCESSOR 00

     Area Under Test:     Lower - [00000000]    Upper - [00000000]

 Diagnostic Control:   Halt on Error [N]   Loop on Test [Y]   List Errors [N]
                       Access Margin [Y]   Refresh Delay = [10] ms
--------------------------------------------------------------------------------
 DIAGNOSTIC TEST NUMBER LIST:

     1 - Basic Memorv Test                 2 - Comprehensive Memory Test
         (Perform Tests 3 and 4)               (Perform Tests 3, 4, 5, and 7)

     3 - Memory Addressing Test            4 - Marching Pattern Test
     5 - Moving Inversions Test            6 - Sliding Diagonals Test
     7 - Refresh Delay Test                8 - Power Fail Test

 ENTER DIAGNOSTIC TEST NUMBER: [1]                           PASSCOUNT = 0000

  (*ERROR INDICATOR)                                     (*DIAGNOSTIC STATE)
--------------------------------------------------------------------------------
        SCREEN CONTROL                               DIAGNOSTIC CONTROL
F2:MESSAGES F3:LOG F8:TERMINATE                  F9:START TEST   F10:STOP
                                              F11:CONTINUE  F16:INITIALIZE
```

T16/81168-01

Figure 3-1 Diagnostic Control Screen

that the diagnostic microcode is waiting for operator instructions. If any noncontiguous memory is encountered during initialization, ERROR INDICATOR also changes, flashing ERRORS DETECTED.

When the test is started, DIAGNOSTIC STATE changes from DIAGNOSTIC IDLE, to flash DIAGNOSTIC EXECUTING. At this time also, the executing memory test is highlighted in DIAGNOSTIC TEST NUMBER LIST by inverse video.

At startup, PASSCOUNT defaults to zero. On completion of a selected test, PASSCOUNT is incremented. The status of the diagnostic then depends on the setting of Loop on Test. If Loop on Test is enabled, DIAGNOSTIC STATE remains unchanged, and the test restarts. If Loop on Test is disabled, DIAGNOSTIC STATE changes from DIAGNOSTIC EXECUTING to DIAGNOSTIC IDLE, and the program waits for further instructions.

If execution is manually interrupted during a test (or if an error is encountered with Halt on Error enabled), DIAGNOSTIC STATE changes from DIAGNOSTIC EXECUTING to DIAGNOSTIC SUSPENDED. If the test is resumed, DIAGNOSTIC STATE returns to DIAGNOSTIC EXECUTING. If an error is detected during execution, ERROR INDICATOR flashes ERRORS DETECTED.

The field at the bottom of the Diagnostic Control Screen summarizes the function keys which can be exercised from this screen. On the left are those for moving to the other available screens; on the right, those for executing the commands specific to the this screen.

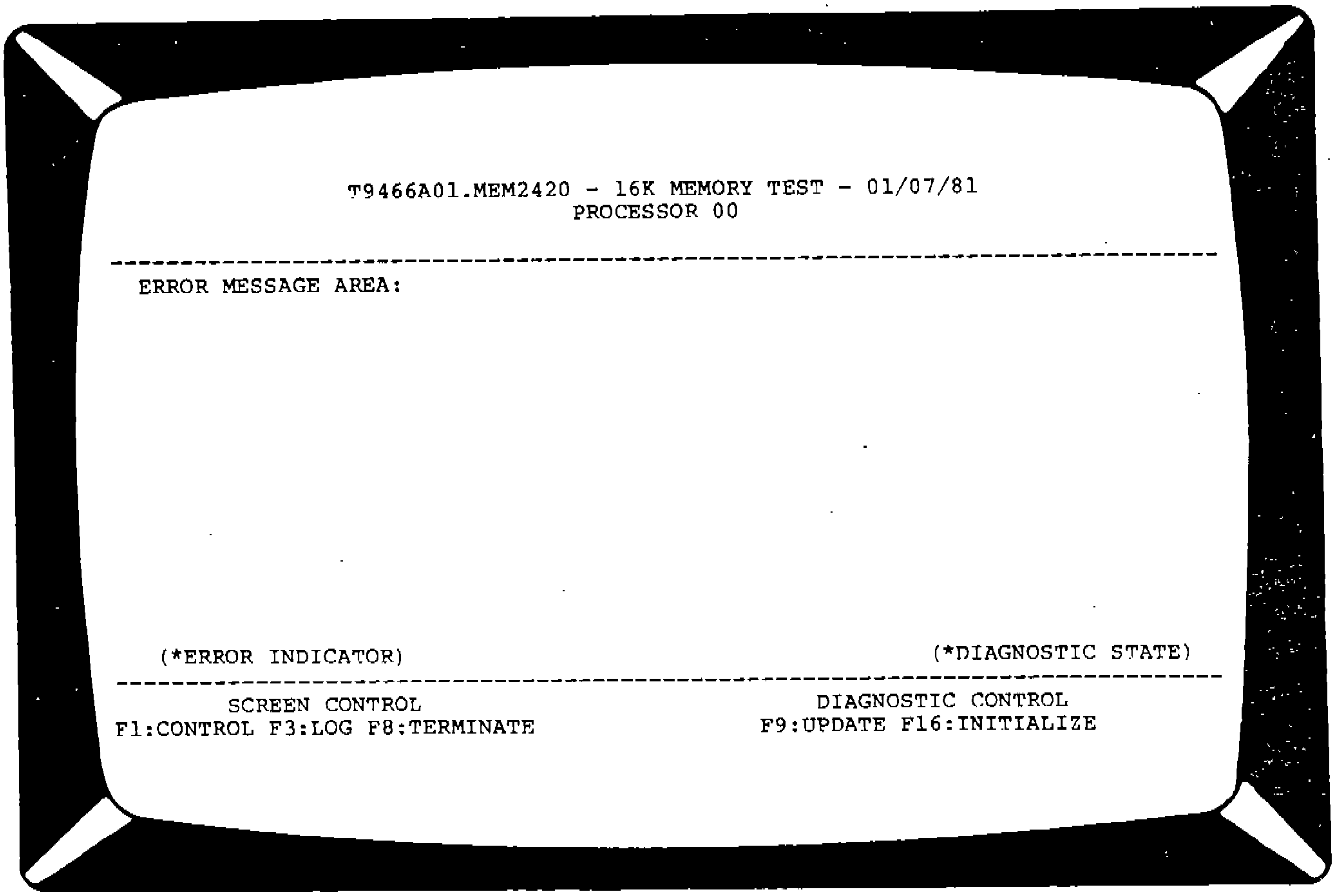
Once the modifiable fields are set as desired, the diagnostic may be started by depressing function key F9. Depressing function key F10 suspends a diagnostic test at any time during its execution. Depressing function key F11 restarts a diagnostic test suspended by a previous F10 command (or one which has encountered an error with the Halt on Error parameter enabled). Depressing function key F16 reinitializes the Memory Diagnostic.

3.1.2 Error Message Screen

The Error Message Screen is invoked by depressing function key F2. The format of the Error Message Screen is shown in Figure 3-2.

At the top of the Error Message Screen is the banner identifying processor number, diagnostic name, and release date. At the bottom is the summary of function keys that can be exercised from this screen.

The remainder of the screen is devoted to ERROR MESSAGE AREA, which displays information about errors encountered. The maximum number of errors that can be displayed at any one time is five. In tests that encounter more than five errors, the screen retains the first four errors encountered. The fifth entry continues to change so that it always records the most recent error encountered. Refer to paragraph 4.0 for details of message interpretation. ERROR MESSAGE AREA also contains the ERROR INDICATOR and DIAGNOSTIC STATE fields, which are updated by the message handler as described in paragraph 3.1.1.



```
                T9466A01.MEM2420 - 16K MEMORY TEST - 01/07/81
                                PROCESSOR 00
---------------------------------------------------------------------------
  ERROR MESSAGE AREA:








   (*ERROR INDICATOR)                                  (*DIAGNOSTIC STATE)
---------------------------------------------------------------------------
         SCREEN CONTROL                           DIAGNOSTIC CONTROL
F1:CONTROL F3:LOG F8:TERMINATE               F9:UPDATE F16:INITIALIZE
```

T16/81168-02

Figure 3-2 Error Message Screen

Two functions are executed from the Error Message Screen. Depressing function key F9 updates the Error Message Screen. Depressing function key F16 reinitializes the Memory Diagnostic.

3.1.3 Error Log Screen

The Error Log Screen is invoked by depressing function key F3. The screen initially displayed is the Board Log Table. The format of this table is shown in Figure 3-3.

The Error Log Screen actually consists of an initial screen and a set of subscreens callable from the initial screen. The initial screen is the Board Log Table. The subscreens are the Bit Log Tables for each board listed in the board log. The Bit Log Table cannot be called up directly from either the Diagnostic Control or the Error Message Screen. It is only available from a previously fetched Board Log Table.

At the top of the Board Log Table is the banner identifying processor number, diagnostic name, and release date. At the bottom is a summary of the function keys that can be exercised from the Board Log Table.

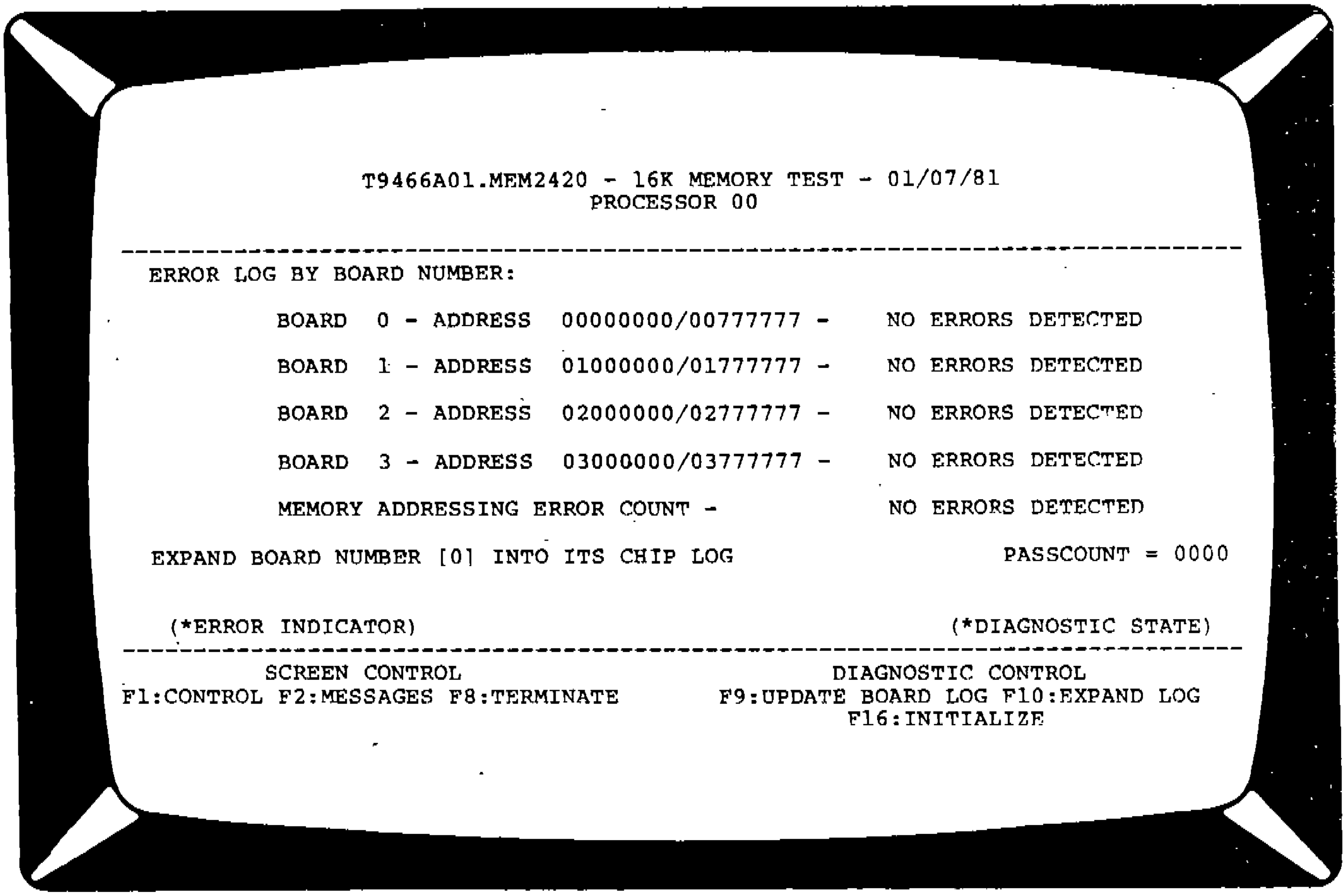
The remainder of the screen is devoted to ERROR LOG BY BOARD NUMBER. This field displays error totals, tabulated for each of the four boards and for memory-addressing type errors, which cannot be isolated to a given board. It also provides the ERROR INDICATOR, DIAGNOSTIC STATE, and PASSCOUNT fields, which are updated by the message handler as described in paragraph 3.1.1, and the special field labeled EXPAND BOARD NUMBER X INTO ITS CHIP LOG.

Any board may be expanded to its bit log by entering the desired board number after EXPAND BOARD NUMBER and depressing function key F10. The format of the Bit Log Table is shown in Figure 3-4.

At the top of the Bit Log Table is the banner identifying processor number, diagnostic name, and release date. The remainder of the screen is devoted to BIT LOG FOR BOARD NUMBER X. This field displays an array corresponding to the chip arrangement of the given memory board.

Depressing function key F9 from either the Board Log or the Bit Log Table updates the error information of the table currently displayed.

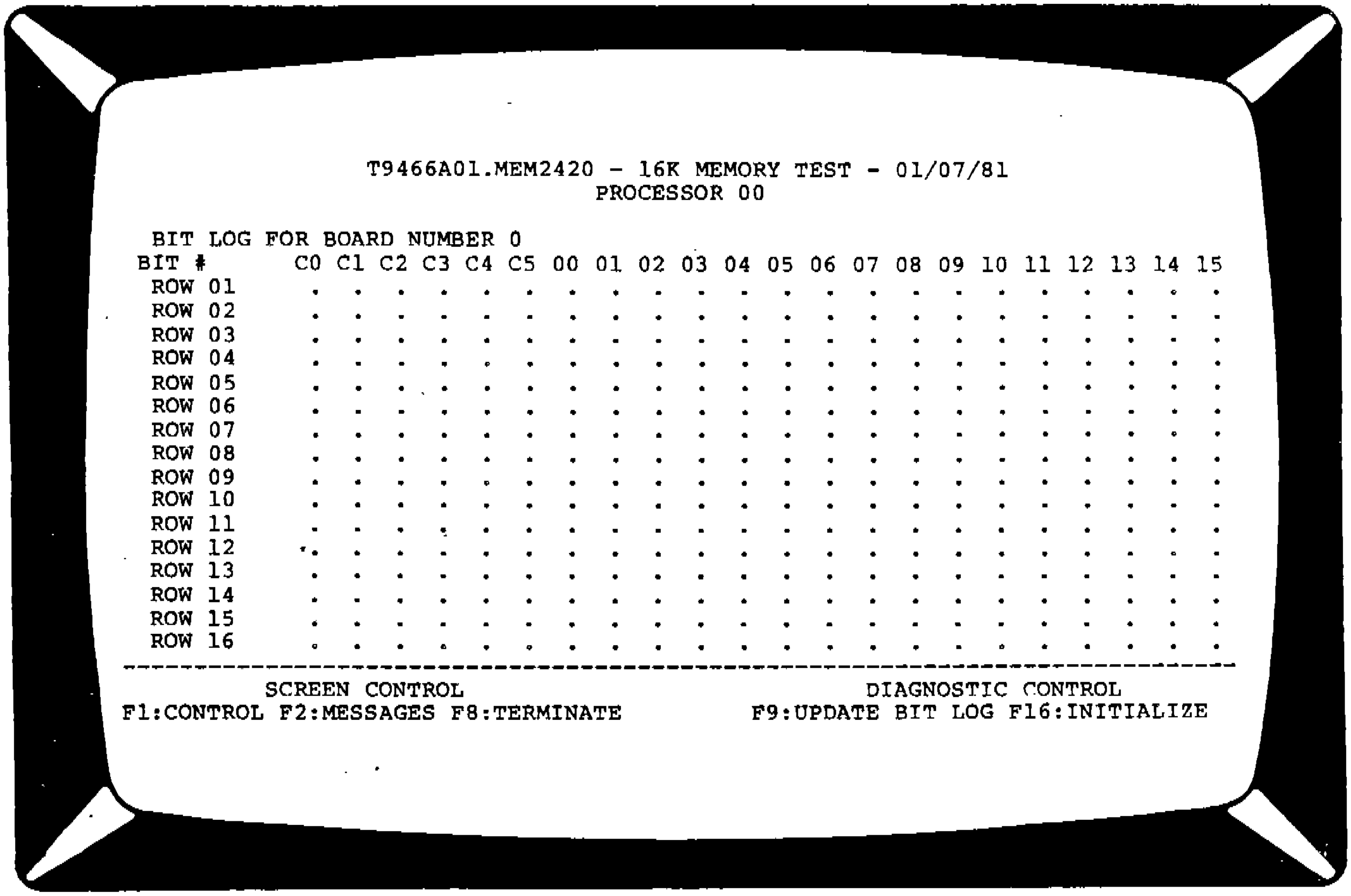
As on the other screens, the command to reinitialize the Memory Diagnostic can be executed from either the Board Log or the Bit Log Table by depressing function key F16.



```
              T9466A01.MEM2420 - 16K MEMORY TEST - 01/07/81
                              PROCESSOR 00

-------------------------------------------------------------------------------
 ERROR LOG BY BOARD NUMBER:

          BOARD  0 - ADDRESS  00000000/00777777 -    NO ERRORS DETECTED

          BOARD  1 - ADDRESS  01000000/01777777 -    NO ERRORS DETECTED

          BOARD  2 - ADDRESS  02000000/02777777 -    NO ERRORS DETECTED

          BOARD  3 - ADDRESS  03000000/03777777 -    NO ERRORS DETECTED

          MEMORY ADDRESSING ERROR COUNT -            NO ERRORS DETECTED

 EXPAND BOARD NUMBER [0] INTO ITS CHIP LOG                  PASSCOUNT = 0000

  (*ERROR INDICATOR)                                     (*DIAGNOSTIC STATE)
-------------------------------------------------------------------------------
          SCREEN CONTROL                           DIAGNOSTIC CONTROL
F1:CONTROL F2:MESSAGES F8:TERMINATE      F9:UPDATE BOARD LOG F10:EXPAND LOG
                                                  F16:INITIALIZE
```

T16/81168-03

Figure 3-3 Board Log Table



```
                   T9466A01.MEM2420 - 16K MEMORY TEST - 01/07/81
                                    PROCESSOR 00

 BIT LOG FOR BOARD NUMBER 0
BIT #       C0 C1 C2 C3 C4 C5 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
 ROW 01      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 02      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 03      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 04      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 05      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 06      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 07      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 08      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 09      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 10      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 11      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 12      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 13      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 14      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 15      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ROW 16      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
-------------------------------------------------------------------------------
          SCREEN CONTROL                                DIAGNOSTIC CONTROL
F1:CONTROL F2:MESSAGES F8:TERMINATE          F9:UPDATE BIT LOG F16:INITIALIZE
```

T16/8116B-04

Figure 3-4 Bit Log Table

3.2 FUNCTION KEY COMMANDS USED IN MEMORY DIAGNOSTIC

A summary of all function key commands used in the Memory Diagnostic is given in Table 3.2.

Table 3.2 Function Key Commands Used in Memory Diagnostic

FUNCTION KEY	OPERATION PERFORMED
1	Fetch Diagnostic Control Screen
2	Fetch Error Message Screen
3	Fetch Error Log Screen
4	Not used in Memory Diagnostic
5	Not used in Memory Diagnostic
6 (shifted)	Fetch Processor Status Screen (used before Memory Diagnostic to select the processor under test)
7 (shifted)	Fetch Microdiagnostic Loader Screen for downloading Memory Diagnostic
8	Terminate diagnostic in currently selected processor; display Control Screen for any other processor running diagnostic (if none, display Processor Status Screen).
9	a. Start selected memory test (from Diagnostic Control Screen) b. Update Error Message Screen (from Error Message Screen) c. Update Board Log Table (from Board Log Table) d. Update Bit Log Table (from Bit Log Table)
10	a. Suspend execution of the test (from Diagnostic Control Screen) b. Expand selected board to its Bit Log (from Board Log Table)

Table 3.2 Function Key Commands Used In Memory Diagnostic (cont'd)

11	Restart a suspended test (from the Diagnostic Control Screen)
12 through 15	Not used in Memory Diagnostic
16	Reinitialize Memory Diagnostic (from Control, Error Message, or Error Log Screen)

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS

Information about errors detected by a memory test is displayed on the Error Message and Error Log Screens. These screens are formatted to facilitate the monitoring and analysis of test results (refer to paragraphs 3.1.2 and 3.1.3).

When an error is encountered, the ERROR INDICATOR field of all screens immediately changes to read ERRORS DETECTED. If Halt on Error is enabled the diagnostic suspends; if not, the diagnostic continues to execute unless manually suspended. Troubleshooting begins at this point. First the Error Message Screen is fetched. This screen identifies the error by one of three numbers indicating its kind. These numbers and their meanings are given in Table 4.1.

Table 4.1 Types of Errors on the Message Screen

NUMBER	MEANING
1	Memory contents miscompare
2	Addressing error or nonexistent memory
3	Noncontiguous memory detected

The screen then gives the failing address, the expected data, and the actual data. The Board Log Table can then be fetched to determine the board on which the failing address resides. Finally, if a board registers errors on the Board Log Table, the Bit Log Table for that board can be fetched to determine the precise chips that are generating the errors.

Even when immediate errors are not generated by a test sequence, it is often advisable to let the Memory Diagnostic loop through a number of passes, in order to detect intermittent or isolated errors. If such errors are occurring, a single pass might not encounter them.

PART 6B - MEMORY DIAGNOSTIC
ERROR CONDITIONS AND DISPOSITIONS

If, during the Memory Diagnostic, a flashing error message appears immediately beneath the banner on any of the three screens, this indicates that a critical error has occurred, not of the kind detected by the Memory Diagnostic, but in the functioning of the OSP itself. Invalid data may have been entered, or an error may have occurred in the Diagnostic Data Transceiver (DDT) communication path. A return to the startup and self-test procedures may be indicated.

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES

To run the Memory Diagnostic, proceed according to the following sequence of steps:

a. Depress shifted function key F6 to select the Processor Status Screen on the OSP. (See Figure 5-1.)

   1. Enter, at the top of the screen, the number of the processor to be tested.

   2. Depress function key F1.

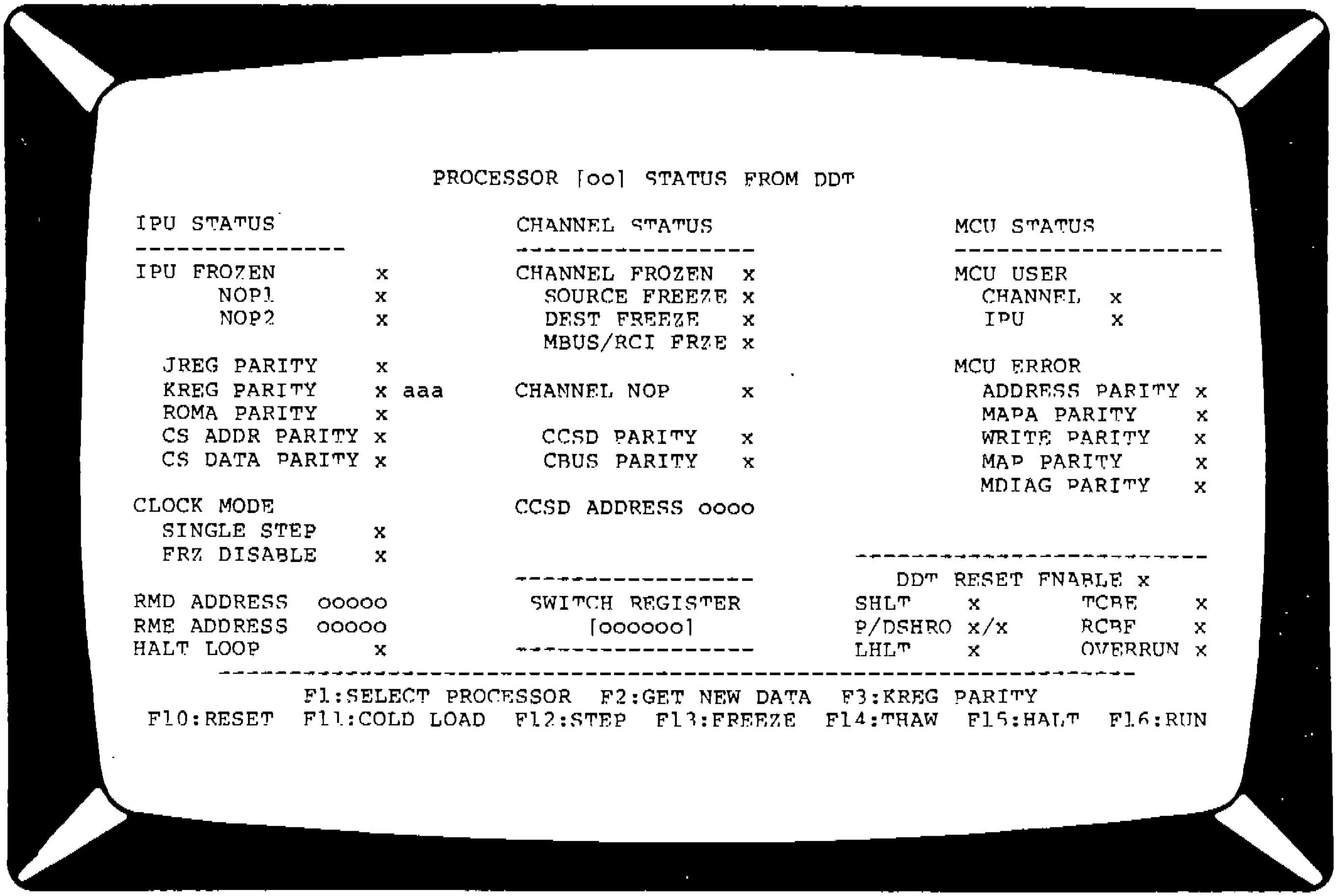
b. Depress shifted function key F7 to fetch the Microdiagnostic Loader Screen on the OSP. (See Figure 5-2.)

   1. Locate the Memory Diagnostic in the list of diagnostic file names on the Microdiagnostic Loader Screen.

   2. Depress the function key whose number corresponds to the number assigned the Memory Diagnostic in the list of file names. (The message handler for the Memory Diagnostic loads into OSP RAM, and the OSP switches to the Diagnostic Control Screen.)

NOTE

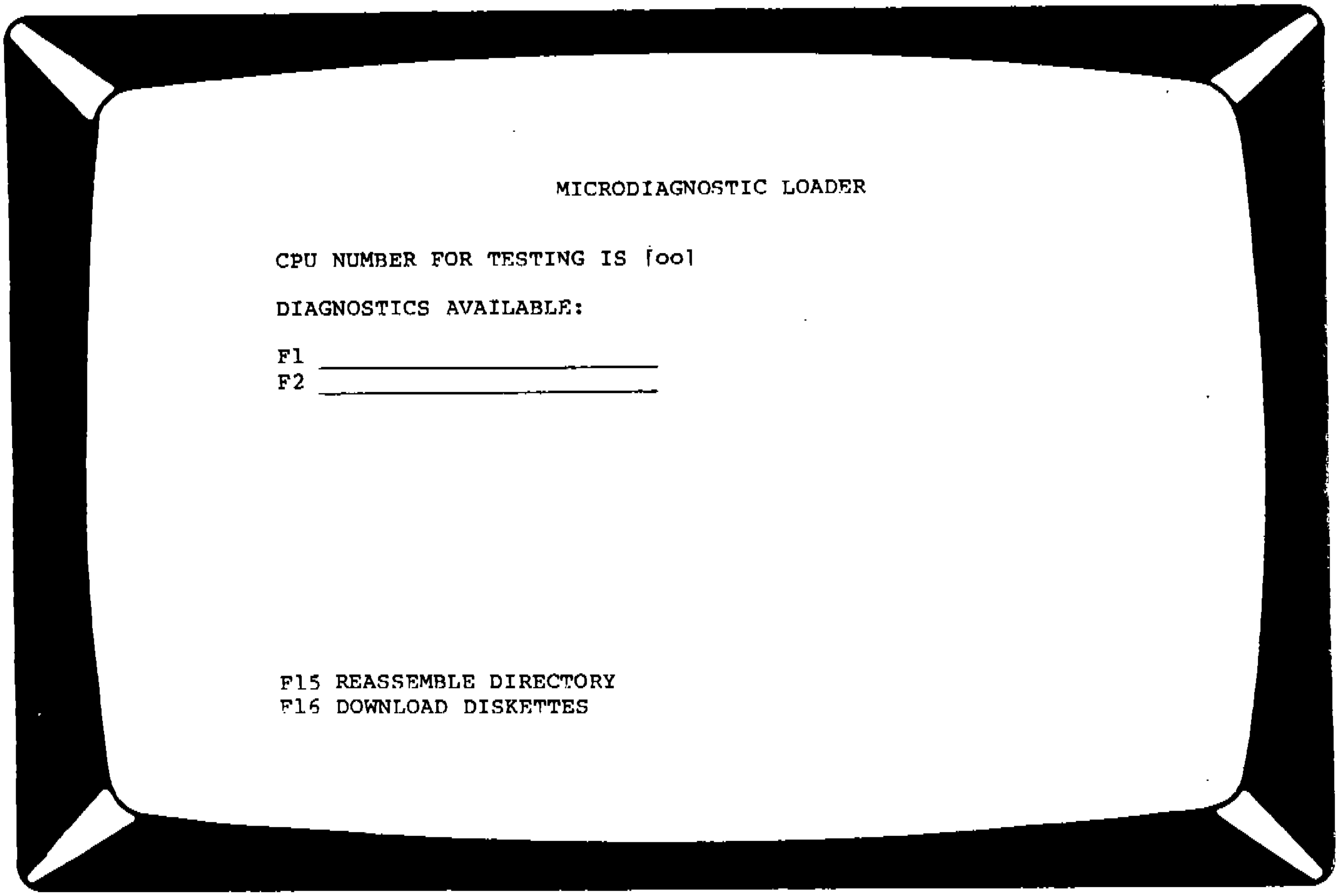
For the diagnostic to be downloaded, the processor under test must be in a halt loop. The message handler accomplishes this by performing a reset of the processor as part of its initialization procedure. However, for the message handler to do this, the RESET ENABLE switch must be set on the Processor Maintenance Interface (PMI) panel. If this switch is disabled when the diagnostic load is attempted, an error message appears on the screen. Refer to paragraph 3.3.3 of the OSP User Guide, 82801-B00, for details of resetting a processor.



```
                         PROCESSOR [oo] STATUS FROM DDT

IPU STATUS                 CHANNEL STATUS                 MCU STATUS
---------------            -----------------              -------------------
IPU FROZEN       x         CHANNEL FROZEN  x              MCU USER
      NOP1       x           SOURCE FREEZE x                CHANNEL   x
      NOP2       x           DEST FREEZE   x                IPU       x
                             MBUS/RCI FRZE x
  JREG PARITY    x                                        MCU ERROR
  KREG PARITY    x aaa     CHANNEL NOP     x                ADDRESS PARITY x
  ROMA PARITY    x                                          MAPA PARITY    x
  CS ADDR PARITY x           CCSD PARITY   x                WRITE PARITY   x
  CS DATA PARITY x           CBUS PARITY   x                MAP PARITY     x
                                                            MDIAG PARITY   x
CLOCK MODE                 CCSD ADDRESS oooo
  SINGLE STEP    x
  FRZ DISABLE    x                                 ------------------------
                           -----------------          DDT RESET ENABLE x
RMD ADDRESS   ooooo          SWITCH REGISTER       SHLT     x       TCBE    x
RME ADDRESS   ooooo             [oooooo]           P/DSHRO  x/x     RCBF    x
HALT LOOP        x         -----------------       LHLT     x       OVERRUN x
      ----------------------------------------------------------------
           F1:SELECT PROCESSOR  F2:GET NEW DATA  F3:KREG PARITY
 F10:RESET  F11:COLD LOAD  F12:STEP  F13:FREEZE  F14:THAW  F15:HALT  F16:RUN
```

T16/81168-05

Figure 5-1 Processor Status Screen



```
                  MICRODIAGNOSTIC LOADER

CPU NUMBER FOR TESTING IS [00]

DIAGNOSTICS AVAILABLE:

F1 ____________________________
F2 ____________________________





F15 REASSEMBLE DIRECTORY
F16 DOWNLOAD DISKETTES
```

T16/8116B-06

Figure 5-2 Microdiagnostic Loader Screen

c. Scan the run-time parameters on the Diagnostic Control Screen.

d. If the default settings of the run-time parameters are not appropriate, change them as required. To change a run-time parameter, proceed as follows:

NOTE

The modifiable parameters are: upper and lower limits of memory, Halt on Error, List Errors, Loop on Test, Access Margin, and Refresh Delay. Refer to paragraph 3.1.1 for a more detailed description of these parameters. To change any parameter once the diagnostic is running, it is necessary to reinitialize the diagnostic.

1. Depress carriage return until the cursor advances to the control field to be changed.

2. Enter the desired change. (The changes are honored by the message handler only when the tests starts.)

   a) For the upper and lower limits of memory, enter the changes as eight-digit octal numbers.

   b) For Halt on Error, List Errors, Loop on Test, and Access Margin, enter N or Y.

NOTE

Enabling Halt on Error automatically enables List Errors, regardless of the setting as displayed or as changed by the operator.

   c) For Refresh Delay, enter the desired millisecond delay as a decimal number.

NOTE

Refresh Delay must be set in millisecond increments and cannot be less than 3 ms or greater than 10 ms.

e. Choose from DIAGNOSTIC TEST NUMBER LIST (on the Diagnostic Control Screen) the test or test sequence to be run.

   1. Depress carriage return until the cursor advances to ENTER DIAGNOSTIC TEST NUMBER.

   2. Enter the number of the desired test after the colon.

NOTE

If Power Fail is the selected test, the procedure differs slightly from that outlined in step f and following. When F9 is depressed to start the Power Fail test, a special instruction is displayed immediately below ENTER DIAGNOSTIC TEST NUMBER on the control screen: "Power processor down and up. Then press F15 to continue." Do as instructed. During the Power Fail test, DIAGNOSTIC STATE registers DIAGNOSTIC IDLE. While the processor is in the down state, the OSP experiences that condition as an error. A 25th line error message is displayed as follows:

CPUXX - DDT failed last transmission

This message is entirely appropriate while the processor is in the down state and does not require special action. The spurious error condition remains until the diagnostic is successfully reloaded as a result of the F15 command.

f. When all modifiable fields are set as desired and the test to be run is selected, depress function key F9 to start the test.

NOTE

The test completes at least one full pass without further necessary intervention, unless Halt on Error is set to Y and an error is detected. In that case, the test suspends execution, DIAGNOSTIC STATE on all screens displays DIAGNOSTIC SUSPENDED, and the message handler waits further instruction. Run time for a full pass varies with the test selected. (Refer to paragraphs 2.5.1 through 2.5.6 for the approximate run-times of the various tests and test sequences.) When a pass completes, PASSCOUNT is updated on the Diagnostic Control and Error Log Screens. If an error is encountered during the pass, ERROR DETECTED flashes on all screens, and--if List Errors is enabled--pertinent information is displayed in the automatic updates of the Error Message Screen. Unless Halt on Error is set to Y and Loop on Test to N, the test continues to loop, updating with appropriate information at the end of each pass.

g. When an error or errors are detected, perform the following:

1. Depress function key F2 to fetch the Error Message Screen. (Refer to paragraph 3.1.2 for a description of this screen.)

2. Depress F9 if an update of currently displayed information is desired. (Unless the diagnostic is suspended, it may be encountering new errors that can be reflected in the F9 update.)

3. Note the number of the indicated error type and locate it in Table 4.1.

NOTE

Error types 1 (memory contents miscompare) and 2 (addressing error or nonexistent memory) are the most likely ones to appear on the Error Message Screen. If the error type is 1, proceed to step h below. Errors of type 2 cannot be isolated to the chip level and therefore are not logged as board or chip errors. A special entry for addressing errors appears in the Board Log Table. If a type-2 error occurs, ensure that legal and existent upper and lower limits of memory are specified on the Diagnostic Control Screen. It is also advisable in the case of type-2 errors to run General Test to determine whether there are Memory Control Board (MCB) errors. General Test can aid in the further resolution of non-memory-related errors. (Refer to Part 6A of this manual.) Error type 3 (noncontiguous memory detected) should only occur during initialization. Errors of this type indicate a problem with the memory boards on a fairly gross scale, such as a memory board or part of a board missing, or a board out of sequence. Such problems must be eliminated before the memory addresses involved can be effectively analyzed for possible errors at the chip level.

h. If the error type is 1, depress function key F3 to fetch the Error Log Screen. (Refer to paragraph 3.1.3 for a description of this screen.)

NOTE

While the Error Message (or Error Log) Screen is being examined, the executing test may or may not be suspended, as desired. If not suspended, the test continues to loop, updating at the end of each pass. To restart a suspended test, depress function key F11 from the Diagnostic Control Screen.

i. Depress F9 if an update of the Error Log Screen is desired.

j. Scan the list of boards to see which ones are reporting errors.

NOTE

Errors of type 2 cannot be isolated to the chip level and therefore are not reported on the Error Log Screen. Thus, it is possible for ERROR INDICATOR to flash ERROR DETECTED even though none of the boards report any error information. Errors of type 2 usually indicate a more serious problem than faulty chips and may require replacement of entire boards. In rare instances, when errors of type 1 occur concurrently with an error of type 2, a triple bit error may be indicated. In such an instance, replacement of the faulty chips should eliminate not only the errors of type 1 but also those of type 2.

k. Expand to the Bit Log Table for the first board reporting any errors:

1. In the EXPAND BOARD NUMBER field, enter the number of the first board reporting errors.

2. Depress function key F10 to call up the Bit Log Table for the indicated board.

NOTE

The Bit Log Table displays an array corresponding to the chip arrangement on the board. The number of errors per board are distributed in this array on a per-chip basis. Thus, the chips that are actually generating the errors can be determined from this screen.

l. Depress function key F9 if an update of the Bit Log Table is desired.

m. Scan the Bit Log Table to locate the chip positions that are returning errors.

n. Replace the indicated faulty chips.

CAUTION

Before removing a memory board to replace chips, power the processor under test off. For details of the chip replacement procedure, refer to section 5 of the Tandem Processor Subsystems Maintenance Manual, 82809.

o. After replacing all faulty chips on the board reporting errors, rerun the Memory Diagnostic to ensure that all failures are eliminated. The rerun procedure is as follows:

1. Power the processor under test back on. (Refer to section 5 of the Processor Subsystems Maintenance Manual, 82809, for details.)

2. Depress F1 (to return to the Diagnostic Control Screen).

3. Depress F16 (to reload and reinitialize the Memory Diagnostic).

4. Reset run-time parameters as desired. (Reinitialization returns all run-time parameters to the default settings.)

5. Depress F9 (to start the diagnostic running).

NOTE

When the processor is powered off to replace the faulty chips, the diagnostic microcode in LCS was lost. In cases where the power is removed temporarily from a processor, F16 not only reinitializes the diagnostic but also reloads the code in LCS.

p. If errors persist, more than one board may initially have had faulty chips. Repeat steps 5.0.g through 5.0.o as many times as necessary to replace the appropriate chips on all boards reporting errors.

q. If all errors cannot effectively be eliminated by the procedure outlined above, report the problem immediately to the Tandem regional support office.

5.1 SPECIAL CONSIDERATIONS

It is possible to run the Memory Diagnostic in a number of processors at the same time. However, each test must be started separately. To do this, observe the following procedure:

a. Start the Memory Diagnostic in the first processor to be tested, as outlined in steps a through f of paragraph 5.0.

b. With the Memory Diagnostic started in this processor, return to the Processor Status Screen by depressing shifted function key F6. Select the next processor to be tested, and repeat steps a through f of paragraph 5.0.

This procedure may be repeated in as many as four processors. Once started, all will continue running until terminated by manual intervention. While several Memory Diagnostics may, in this way, be running simultaneously, only one processor can be actively monitored on the Memory Diagnostic screens at a time. The one being monitored is governed by the processor number currently selected on the Processor Status Screen. For further information, refer to paragraph 3.3.5 of the OSP User Guide, 82801-B00.

It is not possible, on the other hand, to run more than one type of downloaded diagnostic at the same time. Due to space limitations, OSP RAM can accommodate the microcode for only one message handler at a time. Thus, while the message handler for the Memory Diagnostic, once downloaded, can direct the running of the Memory Diagnostic in a number of processors at once, if a different diagnostic (such as General Test) is to be run, the Memory Diagnostic must first be stopped in all the processors where it is running. Only then can the message handler for the new diagnostic be downloaded. In order to prevent an attempt to download a second diagnostic while another diagnostic is already running in one or more processors, the Microdiagnostic Loader Screen is unavailable as long as any diagnostic test is running in any processor. The procedure for terminating multiple tests running simultaneously is as follows:

a. From the Diagnostic Control Screen of the currently monitored processor, perform the following:

   1. Depress function key F16 to reset the processor.

   2. Depress function key F8 to terminate the diagnostic.

NOTE

> If the diagnostic is running in more than one processor, depressing F8 not only terminates the test in the currently selected processor, it also calls up the Diagnostic Control Screen of the next processor in which the diagnostic is running.

b. When the Diagnostic Control Screen of the next processor in which the test is running appears, depress F16 to reset the processor, and then F8 to terminate the diagnostic.

c. Repeat step b until all currently executing diagnostics are terminated.

NOTE

> When the last executing diagnostic is terminated, the final depression of F8 calls up the Processor Status Screen. From this screen, a new processor may then be selected for test, as outlined in paragraph 5.0, step a. The procedure for running a different diagnostic in the newly selected processor is the same as that outlined in paragraph 5.0, step b.

PART 6B - MEMORY DIAGNOSTIC
RUN PROCEDURES

PART 6C

Loadable Control Store

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 1, PART 6C

LOADABLE CONTROL STORE

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

December 1982
Printed in U.S.A.

Tandem Part Number 82803

REVISION RECORD

REVISION	DATE	DESCRIPTION
A	Aug. 1981	First Edition
B	Dec. 1982	Editorial changes.

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 1, CHAPTER 1, PART 6C

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION....................................1-1

SECTION 2
OVERVIEW

2.0 OVERVIEW....................................2-1
2.1 PROGRAM INITIALIZATION....................................2-1
2.2 PROGRAM EXECUTION....................................2-2
2.3 EXAMINATION OF ERROR MESSAGES....................................2-2
2.4 REINITIALIZATION OF THE DIAGNOSTIC....................................2-2

SECTION 3
COMMANDS

3.0 COMMANDS....................................3-1
3.1 SCREEN SELECTION AND MANIPULATION....................................3-1
3.1.1 Diagnostic Control Screen....................................3-2
3.1.2 Error Message Screen....................................3-4
3.1.3 Bit Log Screen....................................3-6
3.2 FUNCTION KEY COMMANDS USED IN LCS DIAGNOSTIC....................................3-6

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS....................................4-1

TABLE OF CONTENTS (Cont'd)

Par.No. Title Page No.

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES....................5-1
5.1 SPECIAL CONSIDERATIONS....................5-6

LIST OF ILLUSTRATIONS

Fig.No. Title Page No.

3-1 Diagnostic Control Screen....................3-3
3-2 Error Message Screen....................3-5
3-3 Bit Log Screen....................3-7

5-1 Processor Status Screen....................5-2
5-2 Microdiagnostic Loader Screen....................5-3

LIST OF TABLES

Table No. Title Page No.

3.1 Screens Available in LCS....................3-1
3.2 Function Key Commands Used in LCS Diagnostic.......3-8

PART 6C - LOADABLE CONTROL STORE DIAGNOSTIC
TABLE OF CONTENTS

B00

PART 6C - LOADABLE CONTROL STORE DIAGNOSTIC
TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The Tandem NonStop II Loadable Control Store diagnostic (T9925A01.LCS420) tests the 8k of RAM contained in Loadable Control Store (LCS).

The LCS diagnostic is a downloadable program. It resides on one of the two floppy discs associated with the Operations and Service Processor (OSP) and is run through the OSP. It consists of the following three tests:

a. Upper 4k RAM test

b. Lower 4k RAM test

c. Entire LCS (8k RAM) test

It may be loaded into the lower 4k of the LCS while the upper half is tested, and then loaded into the upper 4k while testing the lower half. Alternatively, the entire 8k of LCS RAM may be tested at once. The LCS diagnostic automatically relocates.

When the LCS is generating errors, the LCS diagnostic may be run to determine the error type and to isolate the probable area of failure to the chip level. After the faulty chip or chips have been replaced, the diagnostic should be run again as an endtest to ensure that the errors have been eliminated.

The LCS diagnostic is designed to run on all NonStop II operating systems beginning with revision A00.

For a complete description of the OSP, its functions, and its specific relation to the downloadable diagnostics, refer to Tandem OSP User Guide, 82801.

SECTION 2
OVERVIEW

2.0 OVERVIEW

The OSP must be functioning properly in order to run the LCS diagnostic.

Before running the LCS diagnostic, the processor to be tested must first be selected on the Processor Status Screen of the OSP. The LCS diagnostic is then accessed through the Microdiagnostic Loader Screen of the OSP.

Operator interface to the LCS diagnostic occurs through one of three screens generated by the special message handler for this diagnostic. They are displayed at the OSP terminal and manipulated by the use of predefined function keys. These screens are:

a. Diagnostic Control Screen

b. Error Message Screen

c. Bit Log Screen

Each screen is comprised of various control fields (refer to paragraphs 3.1.1 through 3.1.3), modification of which gives the user access to the control functions of the LCS diagnostic.

2.1 PROGRAM INITIALIZATION

When Loadable Control Store is selected from the diagnostics available on the Microdiagnostic Loader Screen, the microcode for its special message handler is immediately downloaded from floppy disc into OSP RAM. The message handler then performs the following functions:

a. Sets up the operating environment.

b. Defaults to the Diagnostic Control Screen.

c. Enters an idle loop and waits for operator instructions.

2.2 PROGRAM EXECUTION

The Diagnostic Control Screen automatically displays information about processor configuration, status of diagnostic execution, and detection of errors in the processor under test. This screen also enables the modification of four run-time parameters: List Errors, Halt on Error, Loop on Test, and the current test selected.

When the fields are modified as desired and the START TEST function key (F9) is depressed, the message handler reads the microcode for the LCS test from floppy disc. Depending on the setting of Loop on Test, the message handler either re-executes the test or enters an idle loop and waits for operator instructions.

If an error is detected during execution of a test, an error handler routine in the diagnostic responds by sending an error message to the message handler, if List Errors is enabled. Errors are logged in a table maintained in the microdiagnostic, and information about them is displayed on the Error Message Screen and Bit Log Screen.

The status of Loadable Control Store after transmission of an error message depends on the setting of the Halt on Error control field. If Halt on Error is disabled, execution of the test resumes at the point of error detection. If Halt on Error is set, the error handler passes control to the command interpreter and suspends the diagnostic.

2.3 EXAMINATION OF ERROR MESSAGES

While the LCS diagnostic is executing, the Error Message Screen can be examined for information about errors being encountered. This screen may be examined while a diagnostic test is running, or the diagnostic test may first be suspended. This choice is entered from the Diagnostic Control Screen.

2.4 REINITIALIZATION OF THE DIAGNOSTIC

The LCS diagnostic can be reinitialized at any time from any of the three screens, by pressing the appropriate function key. Reinitialization terminates the diagnostic being run, clears the Error Message Screen and Bit Log Screen of accumulated information, displays the Diagnostic Control Screen, returns the run-time parameters and passcount to their default settings, and places the message handler in an idle loop, where it waits for operator instructions.

SECTION 3
COMMANDS

3.0 COMMANDS

The LCS diagnostic is accessed and controlled through preformatted screens and predefined function keys on the OSP. Operator interface takes place through a diagnostic message handler that generates the screens and defines the various function keys needed to manipulate them.

3.1 SCREEN SELECTION AND MANIPULATION

At startup, the message handler for the LCS diagnostic defaults to the Diagnostic Control Screen (refer to paragraph 3.1.1). Following startup, any one of the three available screens (refer to Table 3.1) can be fetched from any other, by depressing the function key associated with the desired screen. From each screen, the test operations can be performed by depressing other function keys defined in the message handler.

A special function key (F8) terminates the LCS diagnostic and returns control to the Processor Status Screen. For a description of the use of this function when the LCS diagnostic is running in more than one processor at the same time, refer to paragraph 5.1.

The screens and their associated functions are listed in Table 3.1. The formats of the LCS diagnostic screens and the specific function key commands executable from each are described in paragraphs 3.1.1 through 3.1.3.

Table 3.1 Screens Available in LCS Diagnostic

FUNCTION KEY	SCREEN DISPLAYED
F1	Diagnostic Control Screen
F2	Error Message Screen
F3	Bit Log Screen

3.1.1 Diagnostic Control Screen

The Diagnostic Control Screen is invoked by depressing function key F1. The format of the Diagnostic Control Screen is shown in Figure 3-1.

At the top of the Diagnostic Control Screen is a banner identifying the processor number, test name, and release date. At the bottom is a summary of the relevant function keys and their uses. The remainder of the screen is divided into two active fields: Diagnostic Control and Test Sequence List.

The Diagnostic Control field contains three parameters which can be altered as dictated by the needs of the situation. The default settings are: List Errors set to N (disabled); Halt on Error set to N (disabled); and Loop on Test set to N (disabled).

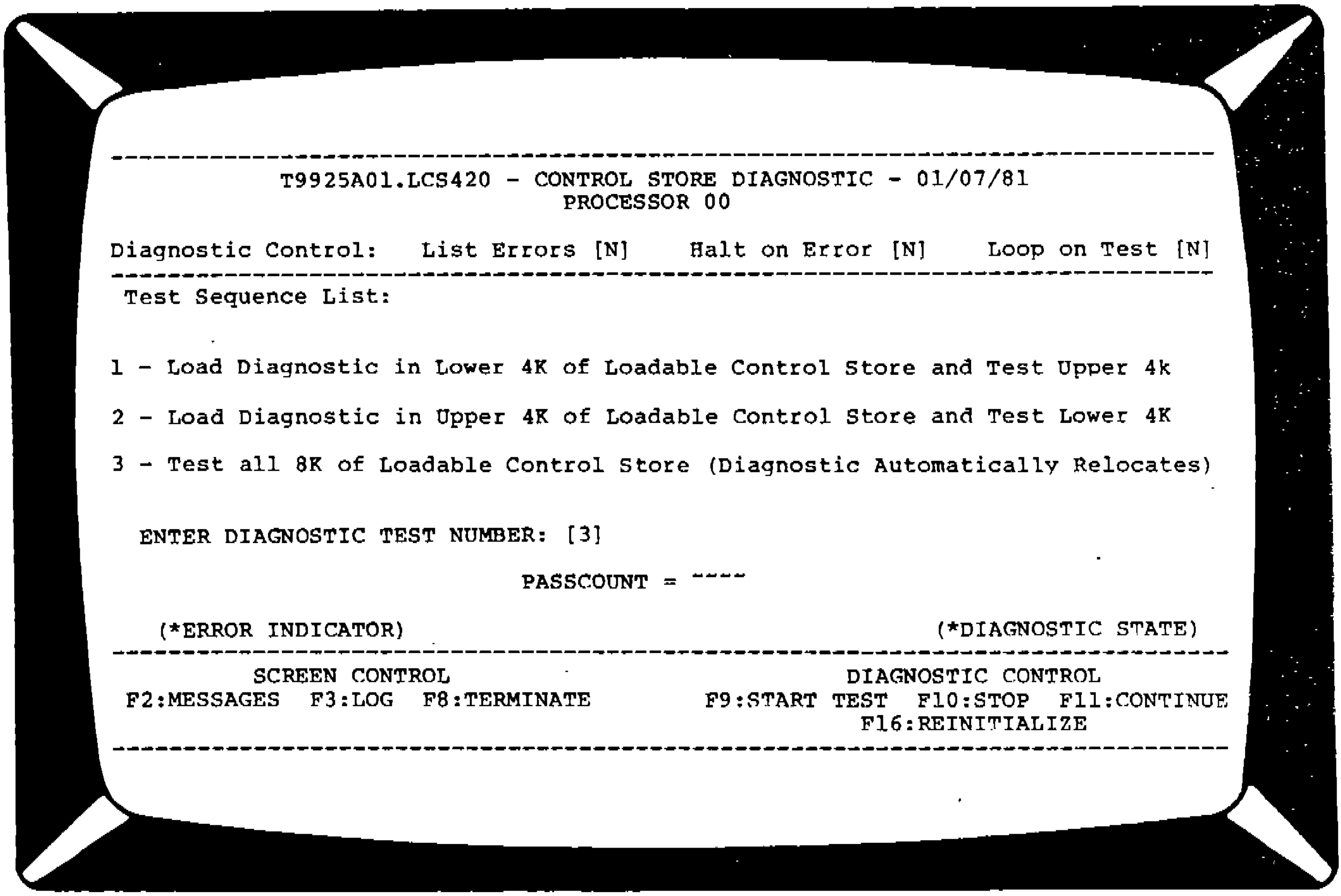
If List Errors is enabled, errors are logged to the Error Message Screen (see figure 3-2). If Halt on Error is enabled, the diagnostic suspends execution if it encounters an error. If Loop on Test is enabled, the LCS diagnostic restarts itself automatically and increments PASSCOUNT by one every time the test sequence is completed.

The area of the Diagnostic Control Screen identified as Test Sequence List enumerates the three test options available in the LCS Diagnostic. The number of the selected test must be entered in the DIAGNOSTIC TEST NUMBER field. It also provides ERROR INDICATOR, DIAGNOSTIC STATE, and PASSCOUNT fields.

ERROR INDICATOR and DIAGNOSTIC STATE are automatically updated by the message handler as the diagnostic runs. When the Loadable Control Store diagnostic is invoked from the Execute Diagnostics Screen and the Diagnostic Control Screen is first displayed, ERROR INDICATOR displays NO ERRORS DETECTED and DIAGNOSTIC STATE displays DIAGNOSTIC IDLE. When function key F9 is depressed to load the selected tests, DIAGNOSTIC STATE flashes DIAGNOSTIC LOADING.

When the test begins to execute, DIAGNOSTIC STATE changes to flash DIAGNOSTIC EXECUTING. The state of The LCS diagnostic on completion of a full pass is governed by the setting of Loop on Test. If Loop on Test is enabled DIAGNOSTIC STATE remains unchanged, PASSCOUNT increments, and the test restarts. If Loop on Test is disabled, DIAGNOSTIC STATE changes from DIAGNOSTIC EXECUTING to DIAGNOSTIC IDLE, and the program waits for further instructions.

If execution is manually interrupted while a diagnostic test is running (or if an error is encountered with Halt on Error enabled), DIAGNOSTIC STATE changes from either DIAGNOSTIC EXECUTING or DIAGNOSTIC LOADING to DIAGNOSTIC SUSPENDED. It changes back again when the test is resumed. If an error is detected during execution, ERROR INDICATOR flashes ERRORS DETECTED.



```
---------------------------------------------------------------------------
               T9925A01.LCS420 - CONTROL STORE DIAGNOSTIC - 01/07/81
                                   PROCESSOR 00

Diagnostic Control:   List Errors [N]    Halt on Error [N]    Loop on Test [N]
---------------------------------------------------------------------------
 Test Sequence List:

1 - Load Diagnostic in Lower 4K of Loadable Control Store and Test Upper 4k

2 - Load Diagnostic in Upper 4K of Loadable Control Store and Test Lower 4K

3 - Test all 8K of Loadable Control Store (Diagnostic Automatically Relocates)


  ENTER DIAGNOSTIC TEST NUMBER: [3]

                                PASSCOUNT = ----

   (*ERROR INDICATOR)                                     (*DIAGNOSTIC STATE)
---------------------------------------------------------------------------
          SCREEN CONTROL                                DIAGNOSTIC CONTROL
 F2:MESSAGES  F3:LOG  F8:TERMINATE        F9:START TEST  F10:STOP  F11:CONTINUE
                                                 F16:REINITIALIZE
---------------------------------------------------------------------------
```

T16/8116C-01

Figure 3-1 Diagnostic Control Screen

The field at the bottom of the Diagnostic Control Screen summarizes the function keys which can be exercised from the screen. On the left are those for moving to the other available screens; on the right, those for executing the commands specific to this screen.

Depressing function key F9 starts the diagnostic running. Depressing F10 suspends a test at any time during its execution. Depressing F11 restarts a test suspended by a previous F10 command (or one which has encountered an error with HALT ON ERROR enabled). Depressing F16 reinitializes the LCS Diagnostic.

3.1.2 Error Message Screen

The Error Message Screen is invoked by depressing function key F2. The format of the Error Message Screen is shown in Figure 3-2.

At the top of the Error Message Screen is the banner identifying processor number, test name, and release date. At the bottom is the summary of function keys that can be exercised from this screen. The remainder of the screen is devoted to the Error Message Area.

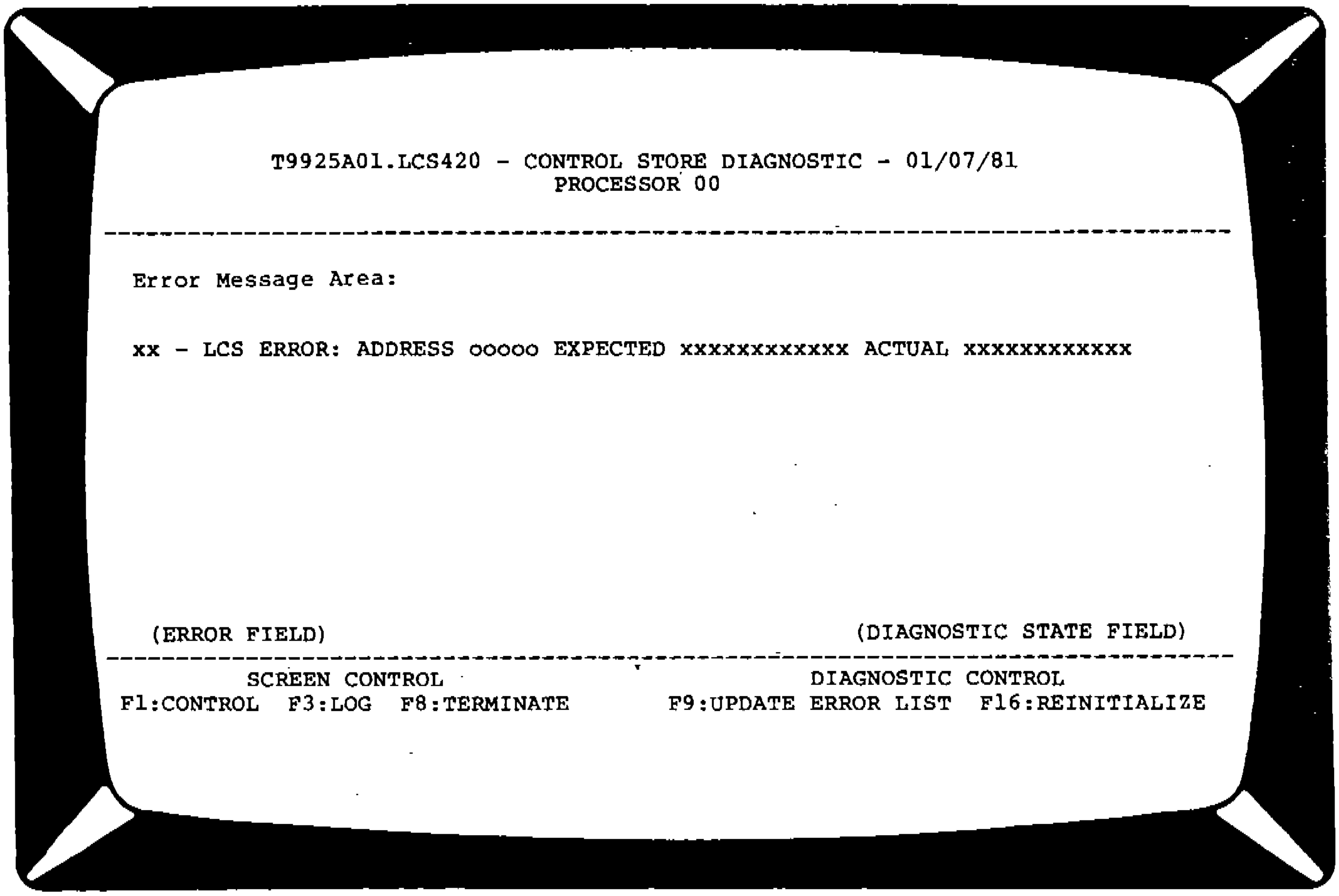
Error Message Area displays information about errors encountered, including the type of error and the error message text.

The maximum number of errors that can be displayed at any one time is seven. In test sequences that encounter more than seven errors, the screen retains the first six errors encountered. The seventh entry continues to change so that it always records the most recent error encountered. For details of message interpretation and fault isolation, refer to paragraph 4.0.

Error Message Area also contains the ERROR INDICATOR and DIAGNOSTIC STATE fields. These are automatically updated by the message handler as described in paragraph 3.1.1.

The Error Message Screen is updated automatically only at the end of every full pass of the LCS diagnostic.This screen can be manually updated during a test sequence by depressing function key F9.

As on the other screens, depressing function key F16 reinitializes the LCS diagnostic.



```
              T9925A01.LCS420 - CONTROL STORE DIAGNOSTIC - 01/07/81
                                  PROCESSOR 00
--------------------------------------------------------------------------------
  Error Message Area:

  xx - LCS ERROR: ADDRESS ooooo EXPECTED xxxxxxxxxxxx ACTUAL xxxxxxxxxxxx






   (ERROR FIELD)                                      (DIAGNOSTIC STATE FIELD)
--------------------------------------------------------------------------------
           SCREEN CONTROL                             DIAGNOSTIC CONTROL
 F1:CONTROL  F3:LOG  F8:TERMINATE       F9:UPDATE ERROR LIST  F16:REINITIALIZE
```

T16/8116C-02

Figure 3-2 Error Message Screen

3.1.3 Bit Log Screen

The Bit Log Screen is invoked by depressing F3. This screen displays the Bit Log Table. The format of this screen is shown in Figure 3-3.

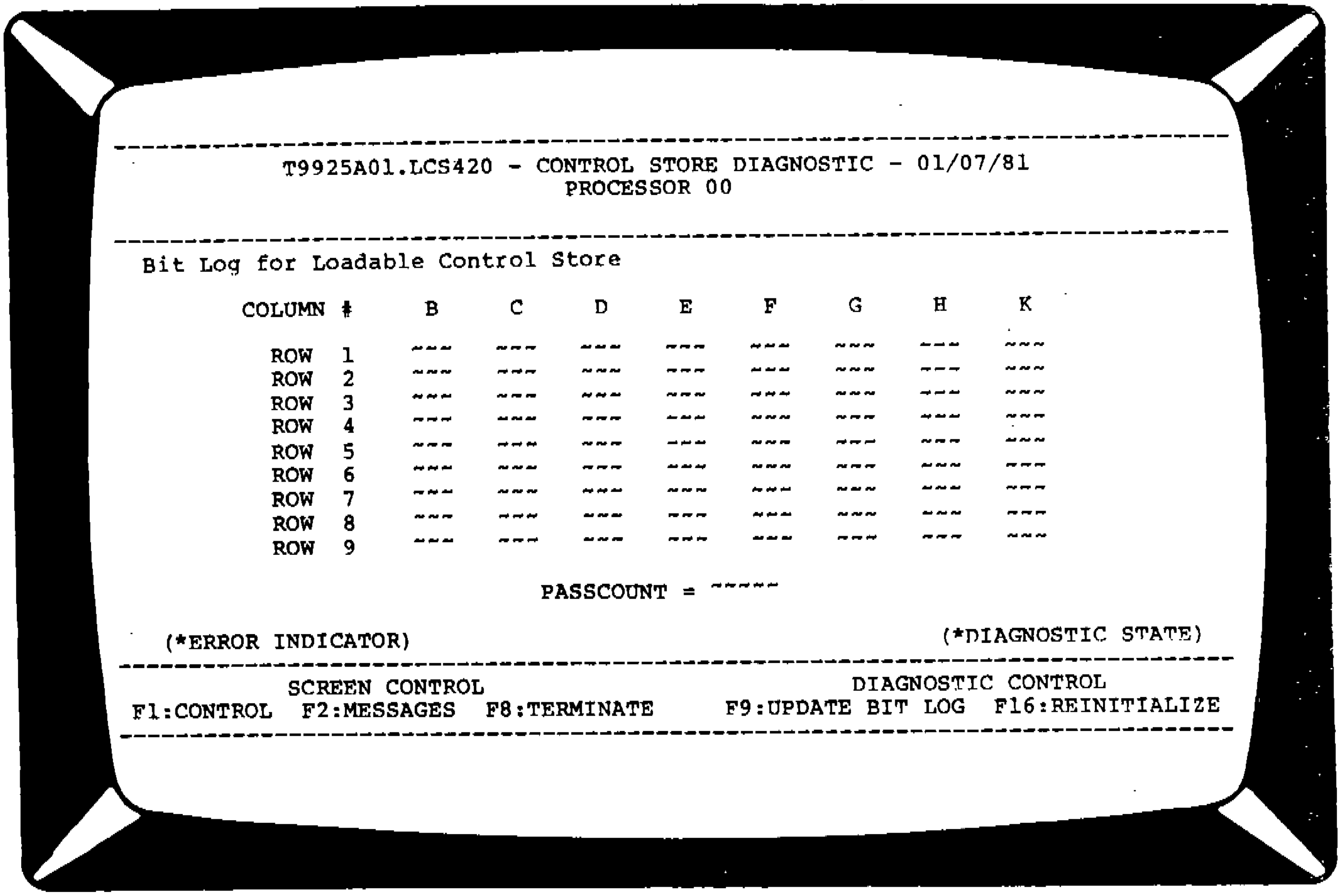
At the top of the Bit Log Table is the banner identifying processor number, diagnostic name, and release date. At the bottom is a summary of the function keys that can be exercised from Bit Log.

The remainder of the screen is devoted to Bit Log for the LCS diagnostic. This field displays an array corresponding to the chip arrangement of the board.

Depressing function key F9 from the Bit Log Table updates the error information of the table currently displayed. The command to reinitialize the diagnostic from the Bit Log Table is executed by depressing function key F16.

3.2 FUNCTION KEY COMMANDS USED IN LCS DIAGNOSTIC

A summary of all function key commands used in the LCS diagnostic is given in Table 3.2.



```
------------------------------------------------------------------------------
          T9925A01.LCS420 - CONTROL STORE DIAGNOSTIC - 01/07/81
                          PROCESSOR 00
------------------------------------------------------------------------------
 Bit Log for Loadable Control Store

        COLUMN #     B     C     D     E     F     G     H     K

          ROW  1    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  2    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  3    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  4    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  5    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  6    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  7    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  8    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~
          ROW  9    ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~   ~~~

                          PASSCOUNT = ~~~~~

  (*ERROR INDICATOR)                                   (*DIAGNOSTIC STATE)
------------------------------------------------------------------------------
           SCREEN CONTROL                          DIAGNOSTIC CONTROL
F1:CONTROL  F2:MESSAGES  F8:TERMINATE    F9:UPDATE BIT LOG  F16:REINITIALIZE
------------------------------------------------------------------------------
```

T16/8116C-03

Figure 3-3 Bit Log Screen

Table 3.2 Function Key Commands Used in LCS Diagnostic

FUNCTION KEY	OPERATION PERFORMED
1	Fetch Diagnostic Control Screen
2	Fetch Error Message Screen
3	Fetch Error Log Screen
4	Not used in LCS diagnostic
5	Not used in LCS diagnostic
6	Return to Processor Status Screen (When LCS diagnostic has been started in a processor and operator wishes to run the diagnostic in a second processor simultaneously)
7	Not used in LCS diagnostic
8	Terminate diagnostic for the current processor; display Control Screen of any other processor running diagnostic (If none, display Processor Status Screen)
9	a. Start LCS diagnostic (from the Diagnostic Control Screen) b. Update error list (from the Error Message Screen) c. Update bit log (from the Bit Log Screen)
10	Suspend execution of the LCS diagnostic (from the Diagnostic Control Screen)
11	Restart a suspended test (from the Diagnostic Control Screen)
12 through 15	Not used in LCS diagnostic
16	Reinitialize LCS diagnostic from the Control, Error Message, or Error Log Screens)

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS

Information about errors detected by the LCS diagnostic is displayed on the Error Message and Bit Log screens. These screens enable the monitoring and analysis of test results and aid in the isolation of failures to the component level. Refer to paragraphs 3.1.2 and 3.1.3 for details about the format of these screens.

When an error is detected, the ERROR INDICATOR field of all screens immediately changes to read ERRORS DETECTED. The problem can then be analyzed by fetching the Error Message Screen. The probable failing chip can be determined from the type of error indicated on that screen.

If a flashing error message appears immediately beneath the banner on any of the three screens, this indicates that a critical error has occurred in the functioning of the OSP itself. Invalid data may have been entered, or an error may have occurred in the Diagnostic Data Transceiver (DDT) communication path. A return to the startup and self-test procedures may be necessary.

The format of the Error Message Screen for the LCS diagnostic is:

"LCS ERROR: ADDRESS XXXXX EXPECTED","XXXXXXXXXX ACTUAL XXXXXXXXX".

The values expressed in this screen are in octal.

PART 6C - LOADABLE CONTROL STORE
ERROR CONDITIONS AND DISPOSITIONS

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES

To run the LCS diagnostic proceed according to the following sequence of steps.

a. Depress shifted function key F6 to select the Processor Status Screen on the OSP. (See Figure 5-1.)

   1. Enter at the top of the screen the number of the processor to be tested.

   2. Depress function key Fl.

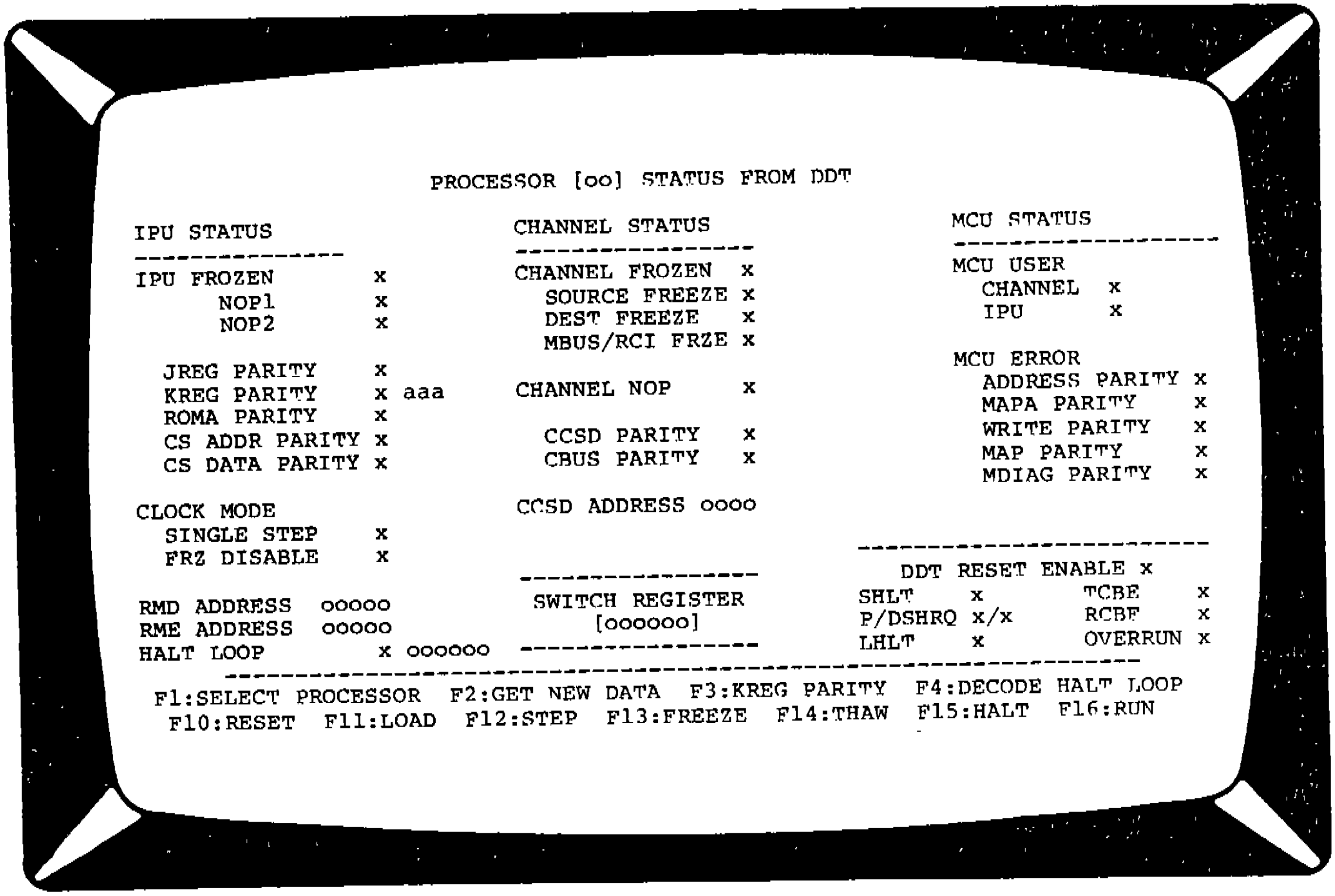
b. Depress shifted function key F7 to fetch the Microdiagnostic Loader Screen on the OSP. (See Figure 5-2.)

   1. Locate Loadable Control Store in the list of diagnostic file names on the Microdiagnostic Loader Screen.

   2. Depress the function key whose number corresponds to the number assigned Loadable Control Store in the list of file names. (The message handler for Loadable Control Store is now downloading, and the OSP switches automatically to the Diagnostic Control Screen.)

NOTE

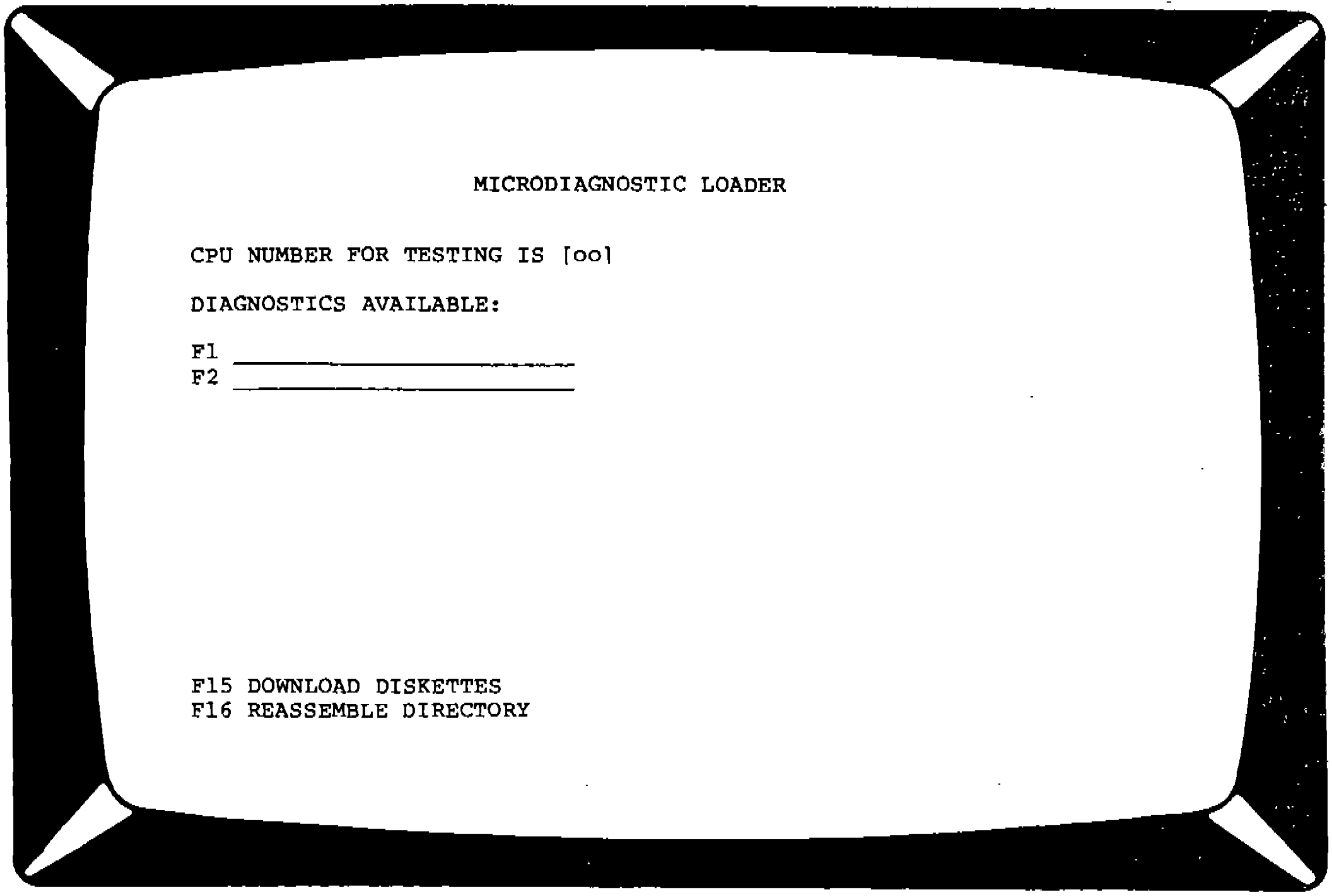
> For the diagnostic to be downloaded, the processor under test must be in a halt loop. The message handler accomplishes this automatically by performing a reset of the processor as part of its initialization procedure. However, for the message handler to do this, the RESET ENABLE switch must be set on the Processor Maintenance Interface (PMI) panel. If this switch is disabled when the diagnostic load is attempted, an error message will appear on the screen.



```
                          PROCESSOR [oo] STATUS FROM DDT

IPU STATUS                    CHANNEL STATUS                MCU STATUS
---------------               ------------------            -------------------
IPU FROZEN          x         CHANNEL FROZEN  x             MCU USER
      NOP1          x           SOURCE FREEZE x               CHANNEL    x
      NOP2          x           DEST FREEZE   x               IPU        x
                                MBUS/RCI FRZE x
  JREG PARITY       x                                       MCU ERROR
  KREG PARITY       x aaa     CHANNEL NOP     x               ADDRESS PARITY x
  ROMA PARITY       x                                         MAPA PARITY    x
  CS ADDR PARITY    x           CCSD PARITY   x               WRITE PARITY   x
  CS DATA PARITY    x           CBUS PARITY   x               MAP PARITY     x
                                                              MDIAG PARITY   x
CLOCK MODE                    CCSD ADDRESS oooo
  SINGLE STEP       x
  FRZ DISABLE       x                                     -------------------------
                              ----------------               DDT RESET ENABLE x
RMD ADDRESS    ooooo           SWITCH REGISTER            SHLT     x       TCBE     x
RME ADDRESS    ooooo              [oooooo]                P/DSHRQ  x/x     RCBF     x
HALT LOOP           x oooooo  ----------------            LHLT     x       OVERRUN  x
     ---------------------------------------------------------------------------
 F1:SELECT PROCESSOR  F2:GET NEW DATA  F3:KREG PARITY  F4:DECODE HALT LOOP
  F10:RESET  F11:LOAD  F12:STEP  F13:FREEZE  F14:THAW  F15:HALT  F16:RUN
```

T16/8116C-04

Figure 5-1 Processor Status Screen



```
                    MICRODIAGNOSTIC LOADER

CPU NUMBER FOR TESTING IS [oo]

DIAGNOSTICS AVAILABLE:

F1 ________________________
F2 ________________________










F15 DOWNLOAD DISKETTES
F16 REASSEMBLE DIRECTORY
```

T16/8116C-05

Figure 5-2 Microdiagnostic Loader Screen

c. Scan the run-time parameters on the Diagnostic Control Screen.

d. If the settings of the run-time parameters, Halt on Error and Loop on Test, are not appropriate, change them as required.

To change a run-time parameter:

1. Depress carriage return until the cursor advances to the control field to be changed.

2. Enter the desired change (N or Y). (The changes are honored by the message handler only when the test starts.)

e. When the modifiable fields are set as desired, depress function key F9 to start the test running.

NOTE

The test completes at least one full pass without further necessary action from the operator, unless Halt on Error is set to Y and an error is detected. In that case, the test suspends execution, DIAGNOSTIC STATE on all screens changes to DIAGNOSTIC SUSPENDED, and the message handler awaits instructions. Run time for a full pass is about five minutes. When a full pass is completed, the test automatically updates Processor Configuration and PASSCOUNT on the Diagnostic Control Screen. If an error is encountered during the pass, ERROR DETECTED flashes on the Diagnostic Control Screen, and pertinent information is displayed in the automatic updates of the Error Message and Bit Log screens. Unless Halt on Error is set to Y and Loop on Test to N, the test continues to loop and updates with appropriate information at the end of each pass.

f. When an error or errors have been detected, perform the following:

   1. Depress function key F2 to fetch the Error Message Screen. (Refer to paragraph 3.1.2 for a description of this screen.)

   2. Depress F9 to update the screen. (The probable area of failure can be determined from the type or types of errors indicated on this screen.)

g. Depress function key F3 to fetch the Bit Log Screen.

NOTE

The Bit Log Table displays an array corresponding to the chip arrangement on the board. The number of errors per board are distributed in this array on a per-chip basis. Thus the chips that are actually generating the error can be determined from this screen.

h. Depress function key F9 if an update of the Bit Log Table is desired.

i. Analyze the information from both the Error Message and the Bit Log Screens to decide which chip is the most likely to be faulty.

j. Replace the suspected faulty chip.

CAUTION

Before removing the CCD to replace chips, power off the processor under test. For details of the chip replacement procedure, refer to section 5 of the Tandem Processor Subsystems Maintenance Manual, 82809.

k. After replacing all faulty chips on the board reporting errors, rerun the LCS diagnostic to ensure that all failures are eliminated. The rerun procedure is as follows:

   1. Power the processor under test back on. (Refer to Section 5 of the Processor Subsystems Maintenance Manual, 82809, for details.)

   2. Depress F16 (to reload and reinitialize the LCS diagnostic).

   3. Reset run-time parameters as desired. (Reinitialization returns all run-time parameters to the default settings.)

   4. Depress F9 (to start the diagnostic running).

l. If errors persist after the chips have been replaced, and all procedures described above completed, report the problem immediately to Tandem regional headquarters.

5.1 SPECIAL CONSIDERATIONS

It is possible to run the LCS diagnostic in a number of processors at the same time. However, each test must be started separately. To do this, observe the following procedure:

a. Start the LCS diagnostic in the first processor to be tested as outlined in steps a through e of paragraph 5.0.

b. With the LCS diagnostic started in this processor, return to the PROCESSOR STATUS SCREEN BY DEPRESSING SHIFTED function key F6. Select the next processor to be tested and repeat steps a through e of paragraph 5.0.

This procedure may be repeated for as many as four processors. Once started, all will continue running until terminated by manual intervention. While several copies of the LCS diagnostic may be running simultaneously, only one processor can be actively monitored on the screen at any given time. The one being monitored is governed by the processor number currently selected on the Processor Status Status Screen.

It is not possible to run more than one type of downloaded diagnostic at the same time. Due to space limitations, OSP RAM can accommodate the microcode for only one message handler at a time. Thus, while the message handler for the LCS diagnostic can direct the testing of Loadable Control Store in as many as four processors at once, if a different diagnostic is to be run, the LCS diagnostic must first be stopped in all the processors where it is running. Only then can the message handler for the new diagnostic be downloaded. To prevent an attempted downloading of a second diagnostic while another diagnostic is already running in one or more processors, the Microdiagnostic

Loader Screen is unavailable as long as a diagnostic test is running in any processor. The procedure for terminating multiple tests running simultaneously is as follows:

a. From any of the diagnostic screens, if the microdiagnostic is running, press function key F16. Then depress function key F8 to terminate the diagnostic message handler for the currently selected processor.

NOTE

> If the diagnostic is running in more than one processor, depressing F8 not only terminates the test in the currently selected processor, it also calls up the Diagnostic Control Screen of the next processor in which the diagnostic is running.

b. When the Diagnostic Control Screen of the next processor in which the test is running appears, depress F8 again to terminate that process.

c. Repeat step b until all currently executing diagnostics are terminated.

NOTE

> When the last executing diagnostic is terminated, the final depression of F8 calls up the Processor Status Screen. From this screen, a new processor may then be selected for test, as outlined in paragraph 5.0, step a. The procedure for running a different diagnostic in the newly selected processor is the same as that outlined in paragraph 5.0, step b.

PART 6C - LOADABLE CONTROL STORE RUN PROCEDURES

CHAPTER 2

Diagnostic Procedures

PART 1

Loading SHADOW and Other Standalone Diagnostics

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 2, PART 1

LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

REVISION RECORD

REVISION	DATE	DESCRIPTION
A00	Aug. 1981	First Edition
B00	Dec. 1982	Manual updated to reflect software changes in October 1982 release and to correct errors in first edition. The title this document now bears differs slightly from the title of the A00 release. (It was formerly called "NonStop II Diagnostic Load Procedures.")

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 1, CHAPTER 2, PART 1

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION....................1-1
1.1 LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS..........1-2

SECTION 2
I/O COLD LOAD FROM TAPE

2.0 I/O COLD LOAD FROM TAPE....................2-1

SECTION 3
BUS COLD LOAD FROM TAPE

3.0 BUS COLD LOAD FROM TAPE....................3-1

SECTION 4
ONLINE LOAD FROM TAPE OR DISC

4.0 ONLINE LOAD FROM TAPE OR DISC....................4-1

SECTION 5
ERROR CONDITIONS DURING DIAGNOSTIC LOAD

5.0 ERROR CONDITIONS DURING DIAGNOSTIC LOAD..................5-1

LIST OF ILLUSTRATIONS

Fig.No.	Title	Page No.
2-1	System Resources for I/O Cold Load from Tape	2-2
3-1	System Resources for Bus Cold Load from Tape	3-2
4-1	System Resources for Online Load from Tape or Disc	4-2
5-1	Tape Controller Status Formats - T16/3202	5-3
5-2	Tape Controller Status Formats - T16/3206	5-4

LIST OF TABLES

Table No.	Title	Page No.
2.1	Sample Responses to FILENAME Prompt During I/O Cold Load From Tape	2-5
2.2	Standard SHADOW Switch Register Settings	2-7
3.1	Sample Responses to FILENAME Prompt During Bus Cold Load From Tape	3-6
4.1	Sample Responses to NEXT FILE? Prompt During Online Load From Tape or Disc	4-9
5.1	Error Conditions During Diagnostic Load	5-2

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The Tandem NonStop II (TM) system provides both for diagnostics that run in an online environment and for diagnostics that run in an offline environment. Those that run in an online environment are written in the EXERCISE diagnostic language and run under the GUARDIAN operating system. The procedure for invoking the EXERCISE environment is described in Volume 1, Chapter 2, Part 6 of this manual. Those that run in an offline, standalone environment are, for the most part, written in the SHADOW diagnostic language; SHADOW diagnostics must be run under the SHADOW operating system. For a description of the SHADOW diagnostic language, refer to the Diagnostic Languages Manual, Part Number 82848.

User interface to an executing standalone diagnostic occurs through the OSP subsystem. Although it is possible to run standalone diagnostics in two or more processors concurrently, only one test process can communicate with the OSP at a time. For this reason, no more than one standalone diagnostic process is ordinarily run on a system at any given time.

CAUTION

The following discussion is relevant only to the loading of TAL or SHADOW diagnostic programs from GUARDIAN disc files or from diagnostic cold load tapes created by DIAGBOOT. The loading of microdiagnostics resident in the OSP on floppy disc is an entirely different procedure. For information regarding that procedure, refer to the OSP User Guide, Part Number 82801-B00, paragraph 3.3.5.

PART 1 - LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS INTRODUCTION

1.1 LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS

There are three methods for loading SHADOW (or any other standalone diagnostic) into an offline processor. They are:

a. I/O cold load from tape

b. Bus cold load from tape

c. Bus load online from tape or disc.

All three of these methods require access to the OSP subsystem; before any of them can be implemented, the OSP and PMI switches must be set to specified positions.

All of the methods depend, then, on the bootstrap loaders contained in DIAGBOOT. The first two methods employ DIAGBOOT indirectly in that they require a diagnostic cold load tape created by DIAGBOOT and containing the apppropriate bootstrap loaders (for tape and bus). The last method invokes DIAGBOOT directly, causing a bus bootstrap to be sent over the interprocessor bus to the offline processor. The procedure for making a diagnostic cold load tape is described in Volume 1, Chapter 3, Part 1 of this manual.

In general, the diagnostic load procedure selected depends on the system resources that are available. I/O cold load from tape is at once the most primitive and the most direct; it requires only an offline processor attached to a tape drive and to the OSP subsytem. Often, however, the processor to be used as the test processor is not conveniently attached to a tape drive. In these cases, one of the indirect load procedures must be used.

Bus cold load from tape requires two offline processors connected by the interprocessor bus; one, which will read the cold load tape, must be attached to a tape drive; the other, which is to be loaded as the test processor, must be attached to the OSP subsystem. Because this method requires two offline processors, it is probably the least desirable for implementation at a customer site.

Online load from tape or disc requires two processors connected by the interprocessor bus, one running GUARDIAN and the other offline; the processor running GUARDIAN must have access to a disc or tape containing the diagnostic source code and to the OSP subsystem; the offline processor need only have access to the OSP subsystem.

These three methods are discussed in detail in Sections 2 through 4. Error conditions are discussed in Section 5.

NOTE

The load procedures outlined in Sections 2 through 4 employ the OSP Processor Status Screen to set Switch Register switches, to actuate the RESET switch, and to initiate a LOAD. If desired, the physical switches on the operator control panel of the test processor may be used instead to perform these functions. This option is noted at the relevant points in each load procedure.

SECTION 2
I/O COLD LOAD FROM TAPE

2.0 I/O COLD LOAD FROM TAPE

This section describes the steps for loading a standalone diagnostic into an offline TNS II processor module directly from a connected tape device. In addition to the offline processor and attached tape drive, this method requires access to the OSP subsystem. A diagnostic cold load tape containing the DIAGBOOT bootstrap loader, the object code for SHADOW, and the source code for the individual diagnostic programs to be run must also be available. The minimal system configuration necessary for I/O cold load from tape is illustrated in Figure 2-1. (The procedure for creating a diagnostic cold load tape is described in Volume 1, Chapter 3, Part 1, of this manual.)

Before an individual diagnostic program can be selected from the cold load tape, the processor must be readied for a cold load and the diagnostic bootstrap loaded.

> CAUTION
>
> Before attempting a diagnostic load, disable all processor PMI FREEZE ENABLE switches.

The step-by-step procedure for I/O cold load from tape is as follows:

a. Ensure that the tape drive to be used is physically connected to the processor to be loaded. (Refer to the "CONTROLLERS" paragraph of the TNS II SYSGEN configuration file to determine the controller and processors to which the tape drive is connected.)

b. Mount the diagnostic cold load tape on the tape drive unit, ensuring that the density setting is correct, that the tape advances properly to the load point, and that the unit comes on line.

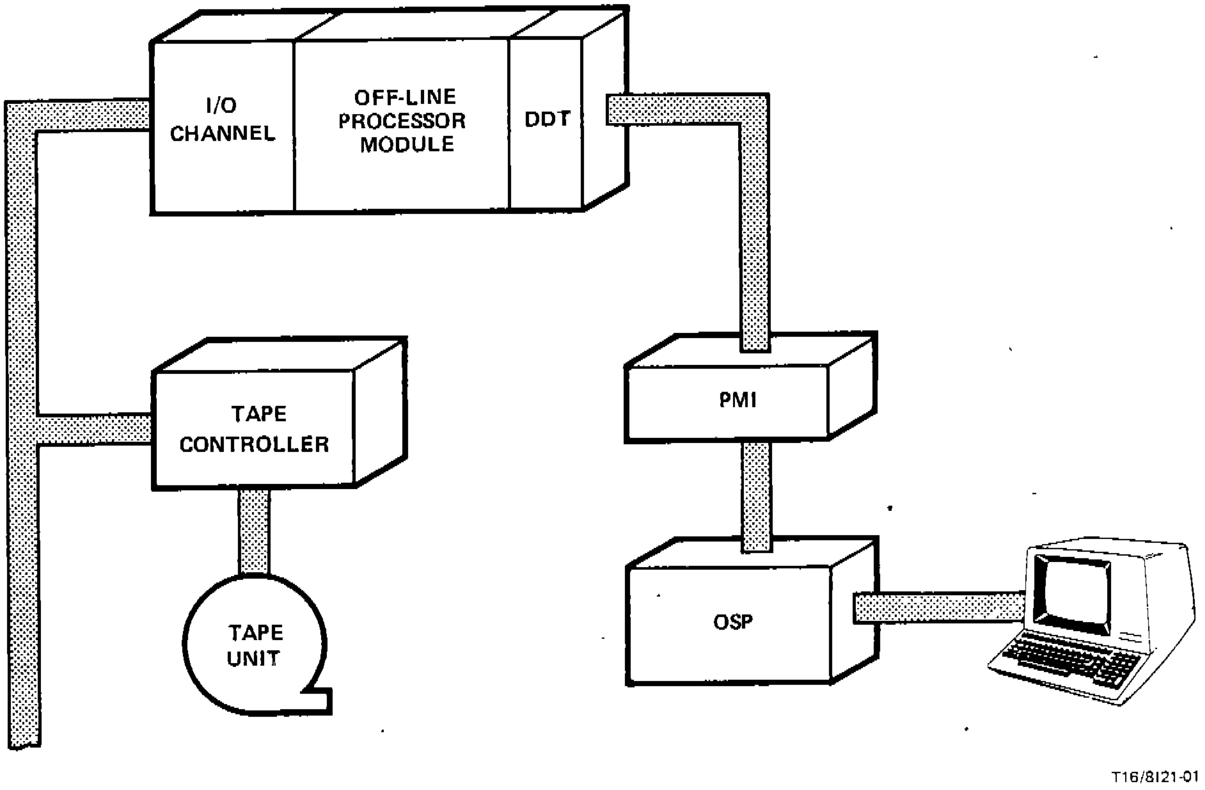


Figure 2-1 System Resources for I/O Cold Load from Tape

c. Set the OSP switches to the following positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to LOCAL

   3. LOCKED/MAINTENANCE to MAINTENANCE

d. Set the PMI switches for the processor to be loaded to the following positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE TO DISABLE

   3. RESET ENABLE TO ENABLE

e. At the OSP terminal, depress function key F6 (shifted) to call up the PROCESSOR STATUS FROM DDT screen. When this screen is displayed, proceed as follows:

   1. Use the TAB key to place the cursor over the Processor Number field, enter (in octal) the number of the processor to be used for the test, and depress function key F1 to transmit the processor number.

   2. Depress function key F10 to RESET the selected processor.

   3. Use the TAB key to place the cursor over the SWITCH REGISTER field, then set the physical address of the tape drive in bits <8> through <15> (controller number in bits <8:12> and unit number in bits <13:15>). Bits <0> through <7> should be set to 0.

   4. Depress function key F11 to initiate the cold load. (The diagnostic bootstrap now loads from the tape.)

NOTE

The procedures described in step d, executed at the PROCESSOR STATUS FROM DDT screen, can be replaced by analogous functions performed at the operator control panel of the processor selected for the test: the address of the tape drive is set into the Switch Register switches, and the RESET/LOAD keyswitch is turned, first to RESET and then to LOAD.

f. Depress function key F2 (shifted) to enter Conversational Mode 2 and wait for the following prompt (from the bootstrap loader) to be displayed:

```
FILENAME?
```

g. At the FILENAME prompt, enter the file name of the standalone diagnostic to be loaded, such as SHADOW or BUS16, and depress carriage return. (Refer to Table 2.1 for examples of this procedure.)

NOTE

> Before requesting that SHADOW (or other standalone diagnostic) be loaded, the user may simply enter a question mark (?) and carriage return at the FILENAME prompt. The bootstrap loader first lists at the OSP terminal the names and types of all the files on the cold load tape and then redisplays the FILENAME prompt (at which time the file name of the standalone diagnostic to be loaded is the appropriate response). If SHADOW is the selected diagnostic, the SHADOW program is loaded, the tape rewinds, and SHADOW enters conversational mode. If an error occurs during the cold load, the processor halts with an error number displayed in the Switch Register lights. Refer to Section 5 for a list of the error codes and corresponding error conditions.

Table 2.1 Sample Responses to FILENAME Prompt During I/O Cold Load from Tape

PROMPT	RESPONSE *	RESULT
FILENAME?	SHADOW <cr>	The SHADOW program is loaded, the tape rewinds, and SHADOW enters conversational mode.
FILENAME?	? <cr>	The names and types of all files on the cold load tape are listed at the terminal, and the FILENAME prompt is redisplayed. Tape searching may be aborted by setting bit <0> of the Switch Register to the ON position until the tape begins to rewind.
FILENAME?	BUS16 <cr>	The program BUS16 is loaded and run.

* If an asterisk (*) is entered following the name of the diagnostic program to be loaded, the program does not begin execution but halts with %177777 displayed in the Switch Register lights. This option is used mainly for software debugging. To begin program execution, depress function key F16 (shifted) at the PROCESSOR STATUS FROM DDT screen for processor being loaded, or toggle the PROGRAM STEP/RUN switch to RUN at the operator control panel of the processor being loaded.

h. Wait for the identifying banner of the specified standalone to be displayed at the OSP terminal (Conversational Mode 2). When SHADOW is the selected program, the form of the banner is as follows:

```
T9404E00.SHADOW  01APR82  00:00
>
```

NOTE

The exact form of this program banner is release-specific. The SHADOW banner is followed immediately by a line feed and the SHADOW prompt (>), which indicates that SHADOW is in conversational mode, waiting for valid user input.

i. When the SHADOW prompt (>) is displayed, proceed as follows:

1. At the operator control panel of the test processor, set the appropriate bits of the Switch Register to the ON position for the desired SHADOW run-control options. The standard Switch Register settings implemented by SHADOW are listed in Table 2.2. Additional switch settings may apply, depending on the specific SHADOW diagnostic program to be run. For the Switch Register options defined for specific SHADOW diagnostics, refer to the descriptions of those diagnostics (elsewhere in this manual).

2. Enter the following command at the SHADOW prompt:

```
RUN <filename>
```

where <filename> is the name of any SHADOW diagnostic program contained on the cold load tape, such as DEX4110, POLL6202, or TAPE3206.

3. Depress carriage return.

NOTE

Any valid SHADOW command may, of course, be entered at the SHADOW prompt. But ordinarily the customer engineer is simply running one of the standard, prewritten SHADOW programs. For a detailed description of the run procedure for the selected diagnostic program, refer to the appropriate discussion elsewhere in this manual.

Table 2.2 Standard SHADOW Switch Register Settings

SWITCH *	FUNCTION
<0>	If toggled on and off, generates a user break request and returns the SHADOW interpreter to conversational mode, where it waits for further instruction.
<3>	If placed in the ON position, suppresses error printout and enables setting of the special numerics ERROR# and COMPERR#.
<4>	If placed in the ON position, suppresses error printout, disables setting of the special numerics ERROR# and COMPERR#, and disables ERROR limit decrement.
<5>	If placed in the ON position, causes a program halt on I/O error.

* Switches <1>, <2>, <6>, and <7> may be defined for specific diagnostics; switches <8> through <15> reflect the I/O address.

B00

SECTION 3
BUS COLD LOAD FROM TAPE

3.0 BUS COLD LOAD FROM TAPE

This section describes the steps for loading a standalone diagnostic into an offline TNS II processor over the interprocessor bus from another offline processor connected to a tape device. This method may be used when no tape drive is conveniently attached to the processor to be loaded as the test processor. The chief drawback of this method is that it requires two offline processors. These processors must be connected by the interprocessor bus: one, which will read the cold load tape, must be attached to a tape drive and to the OSP subsystem; the other, which is to be loaded as the test processor, need only be attached to the OSP subsystem. A diagnostic cold load tape containing the DIAGBOOT bootstrap loader, the object code for SHADOW, and the source code for the individual diagnostic programs to be run must also be available. The minimal system configuration necessary for bus cold load from tape is illustrated in Figure 3-1. (The procedure for creating a diagnostic cold load tape is described in Volume 1, Chapter 3, Part 1, of this manual.)

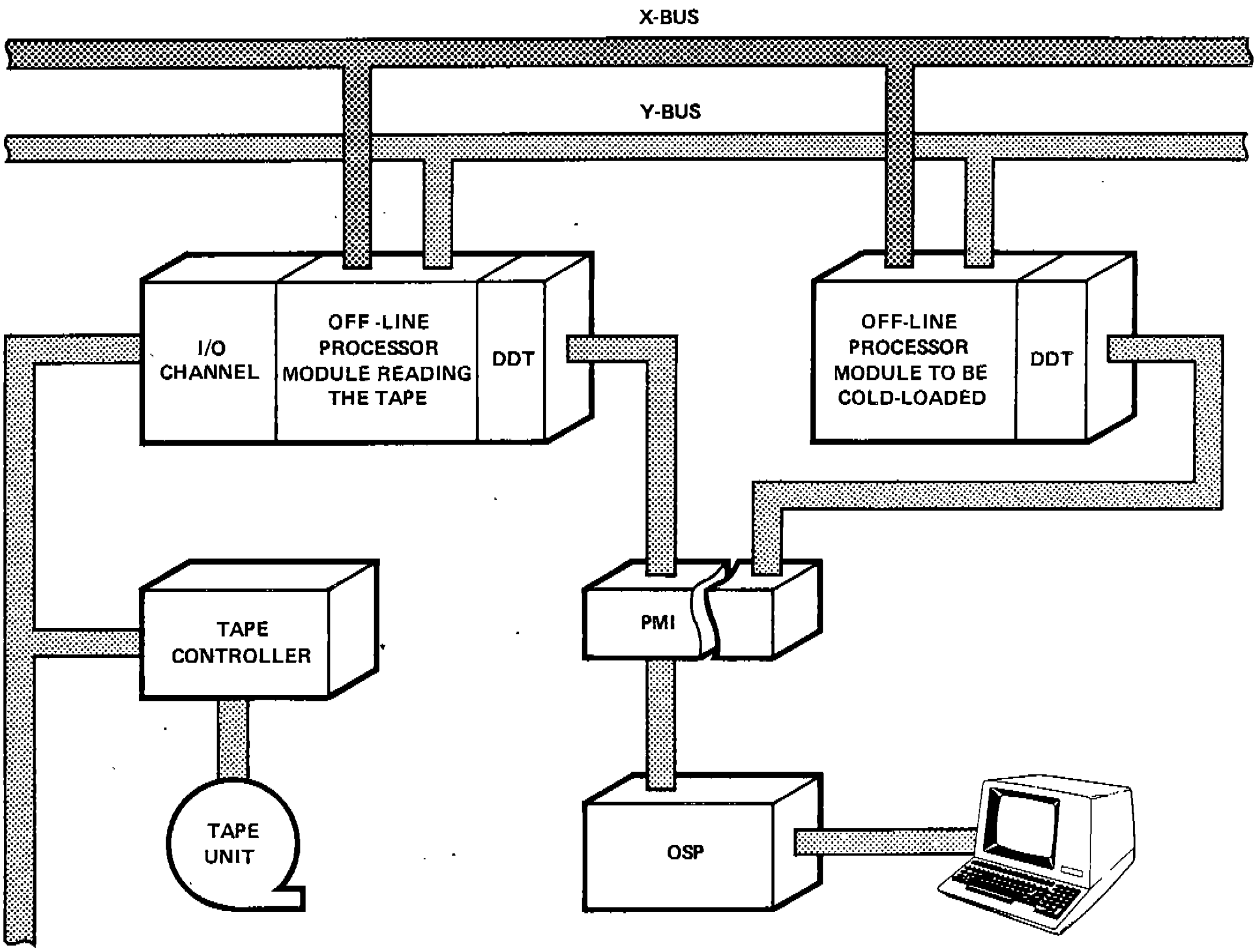
Before an individual diagnostic program can be selected from the cold load tape, the processor must be readied for a cold load and the diagnostic bootstrap loaded.

> CAUTION
>
> Before attempting a diagnostic load, disable all processor PMI FREEZE ENABLE switches.

The step-by-step procedure for bus cold load from tape is as follows:

a. Ensure that the tape drive to be used is physically connected to the processor that is to read and load the tape across the interprocessor bus into the test processor. (Refer to the "CONTROLLERS" paragraph of the TNS II SYSGEN configuration file to determine the controller and processors to which the tape drive is connected.)

b. Mount the diagnostic cold load tape on the tape drive unit, ensuring that the density setting is correct, that the tape advances properly to the load point, and that the unit comes on line.


X-BUS
Y-BUS
I/O CHANNEL
OFF-LINE PROCESSOR MODULE READING THE TAPE
DDT
OFF-LINE PROCESSOR MODULE TO BE COLD-LOADED
DDT
TAPE CONTROLLER
TAPE UNIT
PMI
OSP

T16/8121-02

Figure 3-1 System Resources for Bus Cold Load from Tape

c. Set the OSP switches to the following positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to LOCAL

   3. LOCKED/MAINTENANCE to MAINTENANCE

d. Set the PMI switches for the processor to be loaded to the following positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE TO DISABLE

   3. RESET ENABLE TO ENABLE

e. At the OSP terminal, depress function key F6 (shifted) to call up the PROCESSOR STATUS FROM DDT screen. When this screen is displayed, proceed as follows:

   1. Use the TAB key to place the cursor over the Processor Number field, enter (in octal) the number of the processor to be loaded with the diagnostic (not of the one that is to read the tape), and depress function key F1 to transmit the processor number.

   2. Depress function key F10 to RESET the processor to be loaded with the diagnostic.

   3. Use the TAB key to place the cursor over the SWITCH REGISTER field, then set bit <0> to 1, and bits <1> through <15> to 0.

   4. Depress function key F11 to ready the processor for the cold load.

   5. Use the TAB key to return the cursor to the Processor Number field, then enter (in octal) the number of the processor that is to read the tape and depress function key F1 to transmit the processor number.

   6. Depress function key F10 to RESET the processor reading the tape.

   7. Use the TAB key to place the cursor over the SWITCH REGISTER field, then set the physical address of the tape drive in bits <8> through <15> (controller number in bits <8:12> and unit number in bits <13:15>). Bits <0> through <7> should be set to 0.

8. Depress function key F11 to initiate the cold load. (The processor reading the tape is now loaded with the diagnostic bootstrap.)

NOTE

The procedures described in step e, executed at the PROCESSOR STATUS FROM DDT screen, can be replaced by analogous functions performed at the operator control panels of the processors being used for the test: first, the address of the tape drive is set into the Switch Register switches of the processor to be loaded and the RESET/LOAD keyswitch is turned, first to RESET and then to LOAD. Then, in the Switch Register of the processor reading the tape, bit <0> is set to 1, bits <1> through <11> to 0, and bits <12> through <15> with its CPU number; the RESET/LOAD keyswitch of that processor is then turned, first to RESET and then to LOAD.

f. Depress function key F2 (shifted) to enter Conversational Mode 2 and wait for the following prompt (from the bootstrap loader) to be displayed:

```
FILENAME?
```

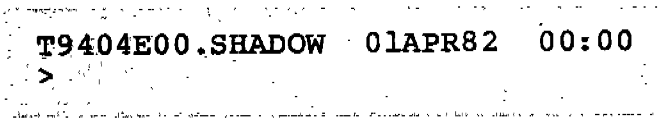
g. At the FILENAME prompt, enter the file name of the standalone diagnostic to be loaded, such as SHADOW or BUS16; the CPU number (in decimal) of the processor to be loaded; and the interprocessor bus to be used (X or Y). Refer to Table 3.1 for examples of this procedure. The syntax of the user input is as follows:

```
<filename>,<cpu><bus>
```

NOTE

Before requesting that SHADOW (or other standalone diagnostic) be loaded, the user may simply enter a question mark (?) and carriage return at the FILENAME prompt. The bootstrap loader first lists at the OSP terminal the names and types of all the files on the cold load tape and then redisplays the FILENAME prompt (at which time "<filename>,<cpu><bus>" is the appropriate response). If SHADOW is the selected diagnostic, the SHADOW program is loaded, the tape rewinds, and SHADOW enters conversational mode. If an error occurs during the cold load, the processor halts with an error number displayed in the Switch Register lights. Refer to Section 5 for a list of the error codes and corresponding error conditions.

h. Wait for the identifying banner of the specified standalone to be displayed at the OSP terminal. When SHADOW is the selected program, the form of the banner is as follows:



```
T9404E00.SHADOW  01APR82  00:00
>
```

NOTE

The exact form of this program banner is release-specific. The SHADOW banner is followed immediately by a line feed and the SHADOW prompt (>), which indicates that SHADOW is in conversational mode, waiting for valid user input.

Table 3.1 Sample Responses to FILENAME Prompt During Bus Cold Load from Tape

PROMPT	RESPONSE *	RESULT
FILENAME?	SHADOW,3Y <cr>	The SHADOW program is loaded into processor 3 over the Y bus, the tape rewinds, and SHADOW enters conversational mode.
FILENAME?	? <cr>	The names and types of all files on the cold load tape are listed at the terminal, and the FILENAME prompt is redisplayed. Tape searching may be aborted by setting bit <0> of the Switch Register to the ON position until the tape begins to rewind.
FILENAME?	BUS16,0X <cr>	The program BUS16 is loaded into processor 0 over the X bus, and run.

* If an asterisk (*) is entered following the name of the diagnostic program to be loaded, the program does not begin execution but halts with %177777 displayed in the Switch Register lights. This option is used mainly for software debugging. To begin program execution, depress function key F16 (shifted) at the PROCESSOR STATUS FROM DDT screen for processor being loaded, or toggle the PROGRAM STEP/RUN switch to RUN at the operator control panel of the processor being loaded.

i. When the SHADOW prompt (>) is displayed, proceed as follows:

1. At the operator control panel of the test processor, set bit <0> of the Switch Register to 0.

2. At the operator control panel of the test processor, set the appropriate bits of the Switch Register to the ON position for the desired SHADOW run-control options. The standard Switch Register settings implemented by SHADOW are listed in Table 2.2. Additional switch settings may apply, depending on the specific SHADOW diagnostic program to be run. For the Switch Register options defined for specific SHADOW diagnostics, refer to the descriptions of those diagnostics (elsewhere in this manual).

3. Enter the following command at the SHADOW prompt:

```
RUN <filename>
```

where <filename> is the name of any SHADOW diagnostic program contained on the cold load tape, such as DEX4110, POLL6202, or TAPE3206.

3. Depress carriage return.

NOTE

The specified program is in turn loaded across the interprocessor bus by the processor that read the tape. Any valid SHADOW command may, of course, be entered at the SHADOW prompt. But ordinarily the customer engineer is simply running one of the standard, prewritten SHADOW programs. For a detailed description of the run procedure for the selected diagnostic program, refer to the appropriate discussion elsewhere in this manual.

PART 1 - LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS
BUS COLD LOAD FROM TAPE

B00

SECTION 4
ONLINE LOAD FROM TAPE OR DISC

4.0 ONLINE LOAD FROM TAPE OR DISC

This section describes the steps for loading a standalone diagnostic into an offline TNS II processor from disc or tape using another processor that is running GUARDIAN. Like bus cold load from tape, this method requires two processors connected by the interprocessor bus. The advantages of online load are that neither processor need be directly attached to a tape drive, and that only the processor to be loaded as the test processor is off line. The other must be running under the GUARDIAN operating system.

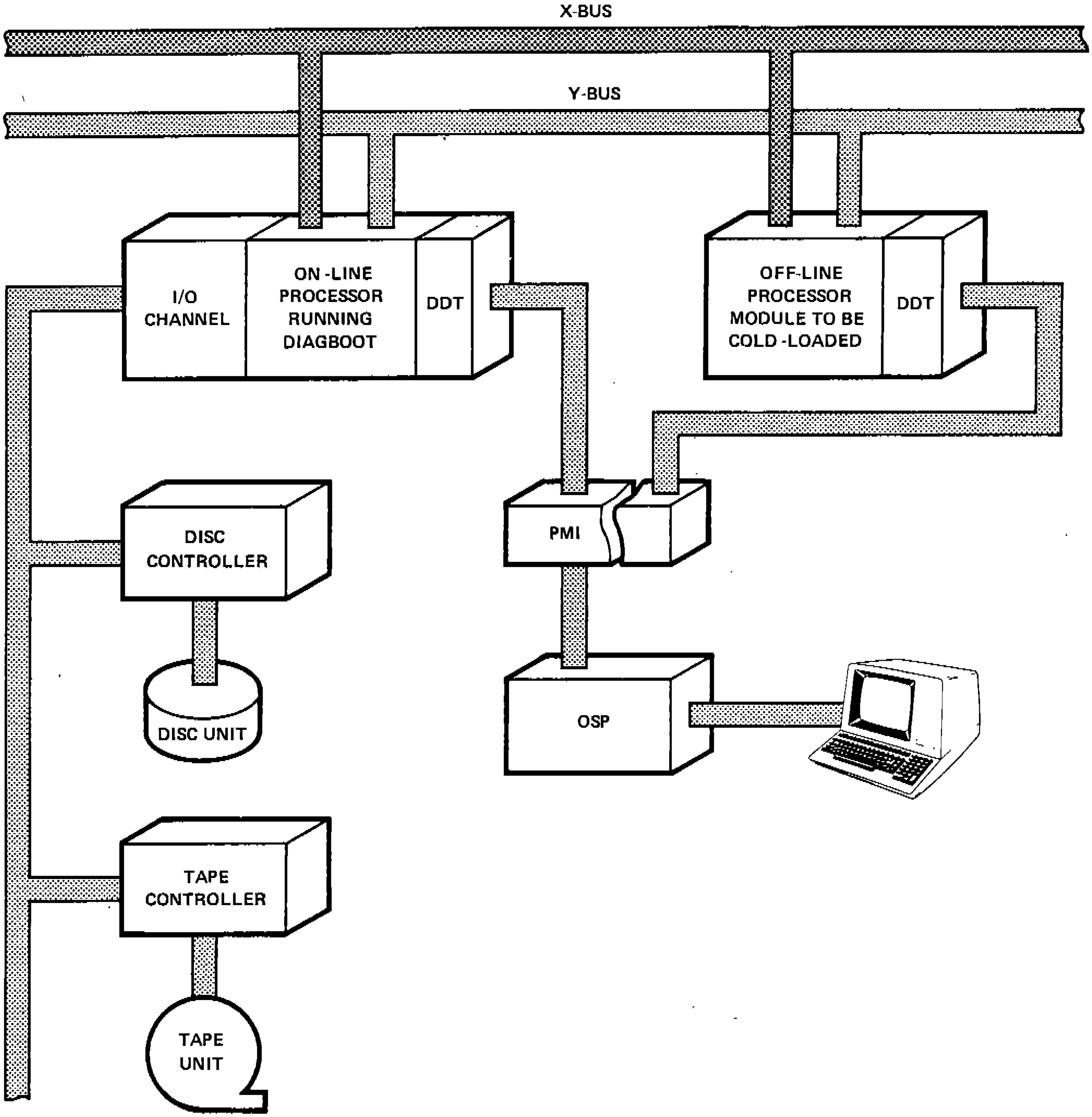
The only complication of online load is the necessity of coordinating inputs on two different screens of the OSP so that the proper sequence is maintained between sending and receiving processors.

The processor running GUARDIAN must have access to the object code for DIAGBOOT and SHADOW and to the source code for the individual diagnostic programs to be run, either on disc or on a diagnostic cold load tape; it must also have access to the OSP subsystem. The offline processor need only have access to the OSP subsystem. The minimal system configuration necessary for online load from tape or disc is illustrated in Figure 4-1. (The procedure for creating a diagnostic cold load tape is described in Volume 1, Chapter 3, Part 1, of this manual.)

The online load procedures from tape and from disc differ in several minor respects. Principally, when the load is from tape, DIAGBOOT issues several preliminary prompts for input, which must be serviced before it issues the prompt for a file to be loaded. When the load is from disc, there are no preliminary prompts; the initial prompt is for a file to be loaded.

NOTE

> When DIAGBOOT is invoked for an online load, the prompt for file to be loaded is NEXT FILE? This differs slightly in form from the corresponding prompt during a cold load. (The prompt during a cold load is FILENAME.)



T16/8121-03

Figure 4-1 System Resources for Online Load from Tape or Disc

Before an individual diagnostic program can be selected from GUARDIAN disc files or from a cold load tape, the processor to be loaded must be readied and the DIAGBOOT program must be run in the online processor.

> CAUTION
>
> Before attempting a diagnostic load, disable all processor PMI FREEZE ENABLE switches.

The step-by-step procedure for online load from tape or disc is as follows:

a. Depending on whether the online load is to be from tape or from disc, perform one of the following two steps:

   1. If the load is to be from tape, mount the diagnostic cold load tape on a tape drive unit accessible to the processor running GUARDIAN, ensuring that the density setting is correct, that the tape advances properly to the load point, and that the unit comes on line.

   2. If the load is from disc, ensure that the disc volume containing the object code for DIAGBOOT and SHADOW and the source code for the individual diagnostic programs to be run is accessible to the processor running GUARDIAN.

b. Set the OSP switches to the following positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to LOCAL

   3. LOCKED/MAINTENANCE to MAINTENANCE

c. Set the PMI switches for the processor to be loaded to the following positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE TO DISABLE

   3. RESET ENABLE TO ENABLE

d. At the OSP terminal, depress function key F6 (shifted) to call up the PROCESSOR STATUS FROM DDT screen. When this screen is displayed, proceed as follows:

   1. Use the TAB key to place the cursor over the Processor Number field, enter (in octal) the number of the processor to be loaded with the diagnostic, and depress function key F1 to transmit the processor number.

   2. Depress function key F10 to RESET the processor to be loaded with the diagnostic.

   3. Use the TAB key to place the cursor over the SWITCH REGISTER field, then set bit <0> to 1, and bits <1> through <15> to 0.

   4. Depress function key F11 to ready the processor for a diagnostic load.

e. Depress function key F1 (shifted) to enter Conversational Mode 1 (A Command Interpeter process is now communicating with the OSP terminal.)

f. At the Command Interpreter (CI) prompt (:) execute one of the following two forms of the command to run DIAGBOOT, depending on whether the load is from tape or disc:

   1. If the online load is from tape, enter the following form of the command at the CI prompt and depress carriage return:

```
DIAGBOOT <device name>
```

   where <device name> is the device name of the tape drive unit, such as $TAPE or $TAPE1. This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to Volume 1, Chapter 3, Part 1, of this manual (paragraph 1.1) for the appropriate syntax.

Respond as follows to the several preliminary requests for input:

a) When the following prompt is displayed:

```
NEW OR OLD?
```

enter the letter O (OLD), indicating that the tape already contains the required diagnostic files, and depress carriage return.

NOTE

The other valid response, N (NEW), is appropriate only if a diagnostic cold load tape is being created. (That procedure is described in Volume 1, Chapter 3, Part 1, of this manual.

b) When the following prompt is displayed:

```
LIST FILES ON TAPE?
```

1) If a list of the files on the tape is desired, enter **Y** at the prompt and depress carriage return.

2) If a list of the files on the tape is not required, enter **N** at the prompt and depress carriage return.

2. If the online load is from disc, enter the following form of the command at the CI prompt and depress carriage return:

```
DIAGBOOT
```

NOTE

This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to Volume 1, Chapter 3, Part 1, of this manual (paragraph 1.1) for the appropriate syntax.

g. When the following prompt is displayed at the OSP terminal (Conversational Mode 1),

```
NEXT FILE?
```

enter the file name of the standalone diagnostic to be loaded, such as SHADOW or BUS16, the CPU number of the processor to be loaded, and the interprocessor bus (X or Y) to be used. Then depress carriage return. Refer to Table 4.1 for examples of this input. The syntax of the user input is as follows:

```
<filename>,<cpu><bus>
```

CAUTION

When loading from disc, it is necessary to specify the exact system location (volume, subvolume, and file) of the standalone diagnostic to be loaded. All diagnostics for the NonStop system II and peripherals are customarily located in the subvolumes designated ZZDIAG or ZZDIAG2. The volume specification, however, may vary. If SHADOW diagnostics are to be run, the SHADOW operating system must be loaded before any individual SHADOW diagnostic may be specified. In this case, SHADOW is the only appropriate response to the first NEXT FILE? prompt (not, for instance, DEX4110 or TAPE3206).

h. When the response to NEXT FILE? is received, DIAGBOOT loads the specified standalone into the test processor, and then displays the following prompt:

```
MICROCODE FILE?
```

Proceed as follows:

1. Respond to the prompt by entering the microcode file name ZZDIAG2.STD15IMG and depressing carriage return.

2. When the DIAGBOOT prompt MICROCODE FILE? is redisplayed, simply depress carriage return. (Ordinarily the only microcode file required is STD15IMG.)

3. When the DIAGBOOT prompt NEXT FILE? is redisplayed, do not respond to it directly. Instead depress function key F2 (shifted) to enter Converstional Mode 2.

i. At the OSP terminal (Conversational Mode 2), wait for the banner of the specified standalone to be displayed. (If an error occurs at the receiving processor, the processor halts with an error number displayed in the Switch Register lights. Refer to Section 5 for a discusssion of error conditions in DIAGBOOT.)

When SHADOW is the selected program, the form of the banner is as follows:

```
T9404E00.SHADOW  01APR82  00:00
>
```

NOTE

The exact form of this program banner is release-specific. The SHADOW prompt (>), which immediately follows the banner, indicates that SHADOW is in conversational mode, waiting for valid user input.

j. When the SHADOW prompt (>) is displayed at the OSP terminal (Conversational Mode 2), proceed as follows:

   1. At the operator control panel of the test processor, set bit <0> of the Switch Register to 0.

   2. At the operator control panel of the test processor, set the appropriate bits of the Switch Register to the ON position for the desired SHADOW run-control options. The standard Switch Register settings implemented by SHADOW are listed in Table 2.2. Additional switch settings may apply, depending on the specific SHADOW diagnostic program to be run. For the Switch Register options defined for specific SHADOW diagnostics, refer to the descriptions of those diagnostics.

   3. Enter the following command at the SHADOW prompt:

```
RUN <filename>
```

   where <filename> is the name of any available SHADOW diagnostic program, such as DEX4110, POLL6202, or TAPE3206.

   4. Depress carriage return.

k. Depress function key F1 (shifted) to return to Conversational Mode 1, where the DIAGBOOT process is still running. Proceed as follows:

   1. At the the DIAGBOOT prompt NEXT FILE? enter the same file name as that entered in the SHADOW RUN command (Converstional Mode 2), followed by a comma, the CPU number of the processor to be loaded, and the interprocessor bus to be used. The CPU number and bus must be same as those specified when the SHADOW operating system was loaded initially. Refer to Table 4.1 for examples of this procedure.

   2. Depress carriage return.

l. Depress function key F2 (shifted) to return to Conversational Mode 2, where the SHADOW operating system is still running. Enter the simple command RUN and depress carriage return.

NOTE

This sequence must be strictly observed so that the receiving processor is set up to receive the specified program before the sending processor is instructed to send it. For a detailed description of the run procedure for the selected diagnostic program, refer to the appropriate discussion elsewhere in this manual.

Table 4.1 Sample Responses to NEXT FILE? Prompt During Online Load from Tape or Disc

DIAGBOOT PROMPT	RESPONSE	RESULT
NEXT FILE?	BUS16,0X <cr>	The program BUS16 is loaded into processor 0 over the X bus, and run.
NEXT FILE?	SHADOW,3Y <cr>	The SHADOW program is loaded into processor 3 over the Y bus, and SHADOW enters conversational mode. (If the load is from tape, the tape rewinds.)
NEXT FILE?	DEX4110,3Y <cr>	Valid response only if SHADOW and microcode file STD15IMG are already loaded in CPU 3 and DEX4110 is also specified at the SHADOW prompt (Conversational Mode 2). If these conditions are met, DEX4110 is loaded into CPU 3 over the Y bus.

* If an asterisk (*) is entered following the name of the diagnostic program to be loaded, the program does not begin execution but halts with %177777 displayed in the Switch Register lights. This option is used mainly for software debugging. To begin program execution, depress function key F16 (shifted) at the PROCESSOR STATUS FROM DDT screen for processor being loaded, or toggle the PROGRAM STEP/RUN switch to RUN at the operator control panel of the processor being loaded.

PART 1 - LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS
ONLINE LOAD FROM TAPE OR DISC

SECTION 5
ERROR CONDITIONS DURING DIAGNOSTIC LOAD

5.0 ERROR CONDITIONS DURING DIAGNOSTIC LOAD

If a serious file system error occurs when DIAGBOOT is running online under GUARDIAN, the process terminates and an appropriate GUARDIAN operating system error message is displayed. For a list of these messages, the conditions they indicate, and the corrective action (if any) to be taken, refer to the GUARDIAN Operating System Messages Manual, Part Number 82076. Other self-explanatory messages may be displayed as DIAGBOOT runs; these are chiefly of the kind originating from operator entry error, which may or may not cause the program to abort.

If an error occurs during an offline load procedure (involving a diagnostic cold load tape created by DIAGBOOT), the processor being loaded halts. A binary error code identifying the condition causing the halt is displayed in the Switch Register lights of the processor. These error codes are listed (in octal) in Table 5.1. They fall into two groups: the first group (negative numbers) occur during the initial phase of the bootstrap load; the second group (positive numbers) occur at any other time during the bootstrap load. In general, errors during the initial phase suggest a malfunctioning tape drive subsystem. Errors after the initial phase are less easily assigned a probable cause. The tape drive subsystem itself is obviously less suspect since it functioned without error during the initial phase. Errors %1 through %7 may indicate a defective tape, while errors %13 and %14 suggest a malfunctioning terminal. Errors %10, %12, and %33 are unlikely occurrences at a customer site; they suggest a program logic error, which may require debugging. The best corrective action in most cases is a simple retry of the load procedure, from a different tape drive or terminal if the error condition suggests it, or with an alternate tape if available. If the halt reoccurs, the error code should be noted and the incident reported.

When a halt is caused by the failure of an I/O operation, certain status information is saved in specific internal registers of the processor. If an EIO instruction fails, the device status is saved in register 0. If an IIO instruction fails, the interrupt cause is saved in register 0; the interrupt status, in register 1. The formats of the EIO and IIO status and cause words for the T16/3202 Tape Controller are shown in Figure 5-1; those for the T16/3206 Tape Controller, in Figure 5-2.

Table 5.1 Error Conditions During Diagnostic Load

SWITCH REGISTER (octal)	ERROR CONDITION
%177776 (-2)	Condition code (CC) after EIO was CCL or CCG
%177775 (-3)	IIO error
%177773 (-5)	Condition code (CC) after IIO was CCG
%177772 (-6)	Ten (10) successive IIO errors on attempt to read record
%177771 (-7)	Attempt to backspace for retry had CCL or CCG after IIO
%177767 (-9)	Initial cold load EIO had CCL or CCG
%000001	Condition code (CC) after EIO was CCL or CCG
%000003	Attempt to backspace for retry had CCL or CCG after EIO
%000004	Attempt to backspace for retry had CCL or CCG after IIO
%000005	Condition code (CC) after IIO was CCG
%000006	10 successive IIO errors on attempt to read record
%000007	Attempt to rewind had CCL or CCG after IIO
%000010	Unexpected interrupt
%000011	Uncorrectable memory error
%000012	Not enough memory to load program
%000013	EIO error on status or write to terminal
%000014	EIO error on read to terminal
%000033	Initial loader tried to load NonStop II System tape bootstrap, but loaded NonStop System tape bootstrap instead
%000070	Too many characters (>28) entered on terminal
%100000 *	Ownership error
%040000 *	Interrupt pending
%020000 *	Channel abort
%010000 *	Parity error

* The interrrupt status errors (also decodable from bits <0> through <3> of the Interrupt Status Word) may occur in combination. The range of possible register displays for these errors is therefore %100000 to %170000. The occurrence of one or more interrupt status errors suggests a malfunctioning tape controller.

EIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	B	P	OV	DPE	CRC	IL	WPV	CR	DNR	TB	EOF	EOT	WR	BOT

O	OWNERSHIP ERROR	WPV	WRITE PROTECT VIOLATION
I	INTERRUPT PENDING	CR	COMMAND REJECT
B	BUSY	DNR	DRIVE NOT READY
P	CHANNEL PARITY ERROR	TB	TAPE BUSY
OV	OVERRUN	EOF	END OF FILE
DPE	DATA PARITY ERROR	EOT	END OF TAPE
CRC	CRC ERROR	WR	WRITE RING
IL	INCORRECT LENGTH	BOT	BEGINNING OF TAPE

Interrupt Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	A	P	0	0	0	0	CONTROLLER NUMBER					UNIT NUMBER		

O	OWNERSHIP ERROR
I	INTERRUPT PENDING
A	ABORT
P	CHANNEL PARITY ERROR

Interrupt Cause Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PON	ATT	RT	UE	OV	DPE	CRC	IL	WPV	CR	DNR	TB	EOF	EOT	WR	BOT

PON	POWER ON INTERRUPT	WPV	WRITE PROTECT VIOLATION
ATT	ATTENTION INTERRUPT	CR	COMMAND REJECT
RT	RUNAWAY TAPE	DNR	DRIVE NOT READY
UE	UNUSUAL END	TB	TAPE BUSY
OV	OVERRUN	EOF	END OF FILE
DPE	DATA PARITY ERROR	EOT	END OF TAPE
CRC	CRC ERROR	WR	WRITE RING
IL	INCORRECT LENGTH	BOT	BEGINNING OF TAPE

T16/8121-05

Figure 5-1 Tape Controller Status Formats - T16/3202

EIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	B	P	U											

O = OWNERSHIP ERROR
I = INTERRUPT PENDING
B = CONTROL UNIT BUSY
P = I/O BUS PARITY ERROR
U = WCS UNLOADED

IIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	A	P	RESERVED				CONTROLLER NUMBER					DRIVE NUMBER		

O = OWNERSHIP ERROR
I = INTERRUPT PENDING
A = CHANNEL ABORT
P = PARITY ERROR

T16/8121-06

Figure 5-2 Tape Controller Status Formats - T16/3206

IIO Cause Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PON	BOT	EOF	EOT	ON LINE	WRITE ENABLED	REW	RESERVED		TERM-STAT BITS						

PON = POWER ON
BOT = BEGINNING OF TAPE
EOF = END OF FILE
EOT = END OF TAPE
OL = ON LINE
WE = WRITE ENABLED
R = REWINDING

TERMINATION STATUS (bits <9> through <15>):

35	INCORRECT LENGTH
36	WRITE RETRIED
37	READ RETRIED
38	SKIP RESIDUE
39	UNCORRECTABLE DATA, CORRECTED
40	UNCORRECTABLE DATA ERROR
41	FORMATTER COMMAND REJECT
42	OPERATION ERROR
43	WRITE FAILURE
44	UNDEFINED COMMAND
45	BAD MICROCODE FILE
46	LARGE READ ERROR
47	RUNAWAY TAPE
48	NOT READY
70	Z80 ERROR
71	PARITY ERROR
72	WRITE-READ LOOP
73	REGISTER ERROR
74	BUFFER ERROR
75	COUNTER/TIMER ERROR
86	Z80 FAILURE
87	PARITY FAILURE
88	WRITE-READ LOOP FAILURE
89	REGISTER FAILURE
90	BUFFER FAILURE
91	COUNTER/TIMER FAILURE
102	Z80 PARITY FAILURE
103	CTL FREEZE TIMEOUT
104	SHORT WRITE
105	BAD MEMORY ACCESS
106	OPERATION TIMEOUT
107	OBUS PARITY FAILURE
108	FCU ROM PARITY FAILURE
109	ADAPTER FAILURE

T16/8121-07

Figure 5-2 Tape Controller Status Formats - T16/3206 (Cont'd)

PART 1 -- LOADING SHADOW AND OTHER STANDALONE DIAGNOSTICS
ERROR CONDITIONS DURING DIAGNOSTIC LOAD

B00

PART 2

Taking a Processor Module Off Line

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 2, PART 2

TAKING A PROCESSOR MODULE OFF LINE

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

VOLUME 1, CHAPTER 2, PART 2

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION..1-1
1.1 OVERVIEW..1-1

SECTION 2
RESET PROCEDURES

2.0 RESET PROCEDURES..2-1
2.1 PREPARATIONS FOR TAKING A PROCESSOR MODULE OFF LINE...2-1
2.2 PROCEDURE FOR TAKING A PROCESSOR MODULE OFF LINE.........2-3

LIST OF ILLUSTRATIONS

Fig. No. Title Page No.

2-1 PROCESSOR STATUS FROM DDT Screen After a RESET......2-4

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

In order to run SHADOW or other offline standalone diagnostics, it is always necessary to have a processor that can be taken off line and dedicated to the diagnostic program.

This document describes the procedure for taking a processor module off line.

1.1 OVERVIEW

Normally, the OSP subsystem is used to take a NonStop II processor module off line. If for some reason the OSP subsystem is not to be used, the RESET/LOAD switch on the operator control panel of the processor may simply be switched to the RESET position.

Although the reset procedure is simple, it must not be undertaken on a customer system without forethought. (Refer to paragraph 2.1 for essential preparations in a customer environment.)

When the processor module is acknowledged as offline by the rest of the system, the following message is printed at the system console:

```
40 <timestamp> <sender cpu,pin> PROCESSOR <num> IS DOWN
```

This message indicates that the <sender cpu> failed to receive an "I'm alive" message from PROCESSOR <num> within twice the configured polling interval.

PART 2 - TAKING A PROCESSOR MODULE OFF LINE
INTRODUCTION

SECTION 2
RESET PROCEDURES

2.0 RESET PROCEDURES

Although taking a processor off line is a simple procedure, the impact of doing so must be considered, both in terms of system configuration and in terms of overall system operation.

These considerations are discussed in paragraph 2.1. The procedure for taking a processor off line follows in paragraph 2.2.

The necessary prerun procedures are ordinarily handled by the local system manager or at least with the concurrence of the system manager.

2.1 PREPARATIONS FOR TAKING A PROCESSOR MODULE OFF LINE

When choosing the processor to be taken off line, consider the following factors:

a. So far as possible, the processor should be selected and the timing of the procedure coordinated in such a way that the downtime has minimum impact on the customer.

b. If the diagnostic program is to be I/O loaded directly from tape, the selected processor must be physically connected to the tape drive used for the I/O load.

c. The selected processor must have access to the OSP subsystem.

d. The selected processor must be physically connected to the device to be tested.

Before taking the processor off line, arrange with the system manager to perform the following:

a. Take the necessary steps to ensure that user processes currently running in the selected processor remain operational when it is taken off line. The procedure is as follows:

   1. Execute the COMINT command PPD to determine which processes, if any, are currently running without backup.

   2. If there are any user processes operating without backup in the processor to be taken off line, restart them in a processor that is to remain operational.

NOTE

When a processor is taken off line, all NonStop (TM) processes currently active in that processor automatically revert to the backup processor assigned to them. The COMINT command PPD is described in the GUARDIAN Operating System Command Language and Utilities Manual, Part Number 82073.

b. Transfer to another processor the path preference of all controllers whose preferred path is currently assigned to the processor to be taken off line. The procedure is as follows:

   1. Execute the PUP command LISTDEV to determine which controllers currently have preferred paths assigned in the test processor.

   2. For all affected controllers, execute the PUP command PRIMARY to reassign the preferred path to the backup processor.

   3. Re-execute the PUP command LISTDEV to verify that the switch takes place.

NOTE

The PUP commands LISTDEV and PRIMARY are described in the NonStop II System Operations Manual, Part Number 82075.

2.2 PROCEDURE FOR TAKING A PROCESSOR MODULE OFF LINE

To take a processor module off line, perform the following:

a. Set the OSP switches of the selected processor to the proper positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to appropriate position

   3. LOCKED/MAINTENANCE to MAINTENANCE

b. Set the PMI switches of the selected processor to the proper positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE don´t care

   3. RESET ENABLE to ENABLE

c. Depress F6 (shifted) to call up PROCESSOR STATUS FROM DDT display.

d. Using the TAB key, place the cursor over the processor number field [oo].

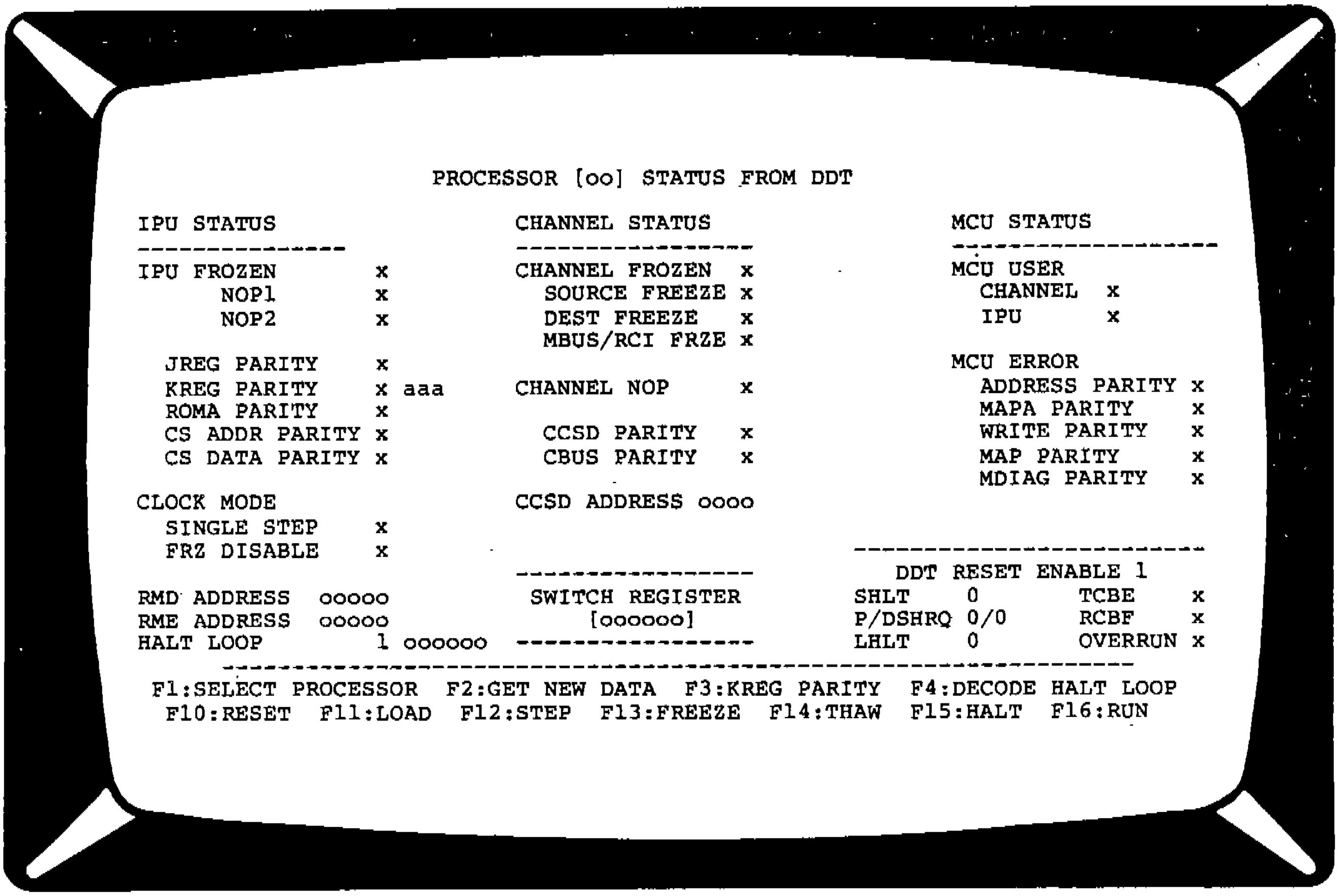
e. Enter (in decimal) the processor number of the selected processor.

f. Depress function key F1 to transmit the processor number.

g. Depress function key F10 to reset the processor. The control panel lights of the reset processor go out, and a message is logged at the system console, acknowledging that the processor is off line.

h. Observe that the HALT LOOP bit is set (1); and that if PSHRQ, LHLT, or DSHRQ were set (1), they are now clear (0). The RESET ENABLE bit is, of course, set (1). Refer to Figure 2-1, where this state of the PROCESSOR STATUS FROM DDT screen is depicted.

> **CAUTION**
>
> Do not RESET a processor running under GUARDIAN unless there is no current SYSTEM HALT REQUEST from the Selected processor or it is acceptable that the halted processors restart at the RESET command! In a system running under GUARDIAN, a processor that is reset must be restarted by doing a RELOAD. Refer to Volume 1, Chapter 2, Part 3, of this manual for a description of the RELOAD procedure.



```
                         PROCESSOR [oo] STATUS FROM DDT

IPU STATUS                    CHANNEL STATUS                  MCU STATUS
---------------               -----------------               -------------------
IPU FROZEN        x           CHANNEL FROZEN  x               MCU USER
      NOP1        x             SOURCE FREEZE x                 CHANNEL   x
      NOP2        x             DEST FREEZE   x                 IPU       x
                                MBUS/RCI FRZE x
  JREG PARITY     x                                           MCU ERROR
  KREG PARITY     x aaa       CHANNEL NOP     x                 ADDRESS PARITY x
  ROMA PARITY     x                                             MAPA PARITY    x
  CS ADDR PARITY  x             CCSD PARITY   x                 WRITE PARITY   x
  CS DATA PARITY  x             CBUS PARITY   x                 MAP PARITY     x
                                                                MDIAG PARITY   x
CLOCK MODE                    CCSD ADDRESS oooo
  SINGLE STEP     x
  FRZ DISABLE     x                                    -------------------------
                              -----------------           DDT RESET ENABLE 1
RMD ADDRESS  ooooo              SWITCH REGISTER        SHLT      0      TCBE     x
RME ADDRESS  ooooo                 [oooooo]            P/DSHRQ 0/0      RCBF     x
HALT LOOP         1 oooooo    -----------------        LHLT      0      OVERRUN  x
   ---------------------------------------------------------------------------
 F1:SELECT PROCESSOR  F2:GET NEW DATA  F3:KREG PARITY  F4:DECODE HALT LOOP
  F10:RESET  F11:LOAD  F12:STEP  F13:FREEZE  F14:THAW  F15:HALT  F16:RUN
```

T16/8122-01

Figure 2-1 PROCESSOR STATUS FROM DDT Screen After a RESET

PART 3

Reloading a Processor Module

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 2, PART 3

RELOADING PROCESSOR MODULES

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

VOLUME 1, CHAPTER 2, PART 3

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION....................................1-1

SECTION 2
RELOAD PROCEDURES

2.0 RELOAD PROCEDURES....................................2-1
2.1 REAPPLYING DC POWER TO A PROCESSOR MODULE................2-1
2.2 LOADING A PROCESSOR MODULE............................2-2
2.3 COLD LOADING THE SYSTEM FROM DISC......................2-3
2.4 ERROR INDICATIONS WHEN COLD LOAD FAILS..................2-7

LIST OF ILLUSTRATIONS

Fig. No. Title Page No.

2-1 PROCESSOR STATUS FROM DDT Screen After Cold Load....2-7

LIST OF TABLES

Table No. Title Page No.

2.1 Cold Load Errors Displayed in the Switch Register Lights....................................2-9

PART 3 - RELOADING PROCESSOR MODULES
TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

This document describes the procedures for reloading processor modules that are off line. The procedure for reloading one or more processors in an otherwise operational system differs from that for cold loading an entire system that is off line.

If one or more processors are taken off line to run diagnostics while the rest of the system remains operational, the offline processors can be reloaded with the Command Interpreter command RELOAD when the diagnostic procedure is complete.

If DC power is removed from a processor while the rest of the system remains operational, care must be exercised when reapplying the DC power. Memory in all cases is invalid, and a processor reload is required.

If a new operating system is installed, or if the entire system is shut down for maintenance purposes, the system can be cold loaded from disc if the disc files containing the operating system image are intact. In system cold load from disc, one processor module is initially cold loaded using the bootstrap load feature. When this processor is successfully loaded, the startup process specified at SYSGEN begins executing. This startup process, which is usually a Command Interpreter, is then used to reload the remaining processor modules in the system.

The step-by-step procedures to be followed in each of these instances are described in paragraphs 2.1 through 2.3.

PART 3 - RELOADING PROCESSOR MODULES
INTRODUCTION

SECTION 2
RELOAD PROCEDURES

2.0 RELOAD PROCEDURES

In some instances a processor to be reloaded will also have been powered off; before such a processor can be reloaded, DC power must be restored to it. The procedure for reapplying DC power to a processor in an otherwise operational system is described in paragraph 2.1. The procedure for reloading a processor in an otherwise operational system follows in paragraph 2.2. And the procedure for cold loading the system from disc is described in paragraph 2.3.

The following information about system configuration is necessary for cold loading that system or for reloading one or more processors in it:

a. For cold loads, the processor on which the cold load is to be done must be physically connected to the disc drive in which the Operating System Image is stored. An appropriate processor may be chosen from the I/O Controller paragraph of the SYSGEN listing.

b. For cold loads, the processor on which the cold load is to be done must be one of the two Operator Processors designated by the SYSGEN as Logical Device $OSP. The numbers of these processors may be determined from the Processors paragraph of the SYSGEN listing.

c. For cold loads, the Subchannel number of the disc (controller and device), which may be determined from the Controllers paragraph of the SYSGEN listing.

d. Both for cold loads and for reloads, the System Subvolume number (SYSxx). This may be determined from the Processors paragraph of the SYSGEN listing.

Notice that items (a) and (b) are simultaneous requirements: the processor chosen must both be connected to the proper disc drive and be one of the Operator Processors that run the OSP Command Interpreter.

2.1 REAPPLYING DC POWER TO A PROCESSOR MODULE

To reapply DC power to a processor module, proceed as follows:

a. Verify that all boards are in the correct slots and are properly seated.

b. Verify that the front edge interconnect cables are installed and properly seated.

c. Locate the DC Power Module, T16/7301 (on the bottom shelf of the processor cabinet, associated from left to right with the four processors in any given cabinet). Proceed as follows:

   1. Set the 115V/230V Power and DC Power switches to the ON position to apply DC power to the processor.

   2. Verify that the 5V/100A, 12V/15A and 5V/18A indicators on the DC Power Module and the POWER indicator on the operator control panel come on and remain lighted.

d. If the processor to which DC power has been restored is to be brought on line, proceed to the steps outlined in paragraph 2.2.

2.2 RELOADING A PROCESSOR MODULE

To reload a NonStop II processor module in a system that is running at least one Command Interpreter and where the OSP subsystem is available, proceed as follows:

a. If DC power has been removed from the processor, follow the steps outlined in paragraph 2.1 before attempting to bring the processor on line. Otherwise, proceed to step b.

b. Add the processor to be reloaded to the Poll List and enable it for RESET:

   1. Set the RESET ENABLE of the appropriate PMI to the ENABLE position.

   2. At the OSP terminal, depress function key F5 (shifted) to call up SYSTEM PROCESSORS STATUS screen.

   3. Using the TAB key, place the cursor over the Poll field [x] of the processor to be reloaded.

   4. Enter a one (1) in the Poll field.

   5. Depress function key F1 to transmit the new Poll List.

c. Make the processor that is to be reloaded the selected processor and reset it:

   1. Depress function key F6 (shifted) to call up the PROCESSOR STATUS FROM DDT screen.

   2. Using the TAB key, place the cursor over the processor number field.

   3. Enter in decimal the processor number of the processor to be reloaded.

   4. Depress function key F1 to transmit the processor number.

   5. Reset the processor by depressing function key F10 (RESET).

d. Load the SWITCH REGISTER:

   1. Using the TAB key, place the cursor over the SWITCH REGISTER field (on the PROCESSOR STATUS FROM DDT screen).

   2. Enter the octal value %100000 in the SWITCH REGISTER field.

e. Reload the processor:

1. Depress function key F11 to LOAD the processor.

2. Place the OSP in Conversational Mode by pressing function key F1 (shifted).

3. At the COMINT prompt (:), enter the following command to load the Operating System Image into the processor:

```
RELOAD <cpu> [,$ <volume name>.SYS<xx>.OSIMAGE] [,NOSWITCH]
                [,<bus>]
```

where <cpu> selects the number of the processor to be reloaded; <volume name> selects the name of the disk volume that contains the Operating System Image for the processor; NOSWITCH, if present, causes the device ownership switches, established by the SYSGEN, to be overridden; and <bus> selects the Interprocessor Bus used during the RELOAD. Zero (0) represents the X bus, and one (1) the Y bus; the default is 0.

For example:

```
:RELOAD 2, $SYSTEM.SYS02.OSIMAGE
```

specifies that CPU 2 be reloaded from $SYSTEM.SYS02.OSIMAGE over the IPB X bus.

f. To check the status of the reloaded processor, call up up the SYSTEM PROCESSORS STATUS screen by depressing function key F5 (shifted).

g. If reloading one or more processors as a function of system cold load from disc, create a backup process for the start-up Command Interpreter by using the BACKUPCPU command:

```
:BACKUPCPU <cpu>
```

where <cpu> is the processor number of the processor in which the backup process is to reside.

2.3 COLD LOADING THE SYSTEM FROM DISC

To cold load the system from disc, proceed as follows:

a. Select a processor module that is physically connected to a disc volume containing the operating system image for the processor. (This information can be found in the I/O Controller Paragraph of the SYSGEN listing for the system.)

b. Set the OSP switches of the selected processor to the proper positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to appropriate position

   3. LOCKED/MAINTENANCE to MAINTENANCE

c. Set the PMI switches of the selected processor to the proper positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE to DISABLE

   3. RESET ENABLE to ENABLE

d. Disable the PMI FREEZE ENABLE switches of ALL other processors.

e. Enable the selected processor for polling:

   1. Depress function key F5 (shifted) to call up SYSTEM PROCESSORS STATUS screen.

   2. Using the TAB key, place the cursor over the Poll field [x] of the selected processor.

   3. Enter a one (1) in the Poll field.

   4. Depress function key F1 to transmit the new Poll List.

f. Make the selected processor the Global Processor Number for OSP operations and RESET the selected processor:

   1. Depress function key F6 (shifted) to call up PROCESSOR STATUS FROM DDT screen

   2. Using the TAB key, place the cursor over the processor number field.

   3. Enter in decimal the number of the selected processor.

   4. Depress function key F1 to transmit the processor number.

   5. Depress function key F10 to RESET the selected processor.

g. Load the SWITCH REGISTER and LOAD the processor:

   1. Using the TAB key, place the cursor over the SWITCH REGISTER field.

   2. Enter in this field the following:

      a) In bits <8:15>, the Subchannel number (octal) of the disc (controller and device).

      b) In bits <1:6>, the octal digits of the System Subvolume. Refer to the SYSGEN listing for the number assignment of the System Subvolume.

      c) In bits <0> and <7>, zeros.

   3. Depress function key F11 to initiate the LOAD. The Operating System loads from Subvolume SYSxx and creates a start-up Command Interpreter for the OSP.

   When the cold load completes sucessfully, the message PROCESSOR UP is displayed at the OSP terminal and the value %177777 is indicated in the processor Switch Register lights (but not in the SWITCH REGISTER field of the PROCESSOR STATUS FROM DDT screen). Refer to paragraph 2.4 for a discussion of the error indications if a cold load fails. Figure 2-1 shows processor 0 being loaded from System Subvolume SYS12; the controller number is %11; the logical device number of the disc drive is %0. The HALT LOOP bit and the RESET ENABLE bit are set (1) since the processor had to be reset.

h. Depress function key F1 (shifted) to enter Conversational Mode. The OSP terminal now operates in Conversational Mode #1.

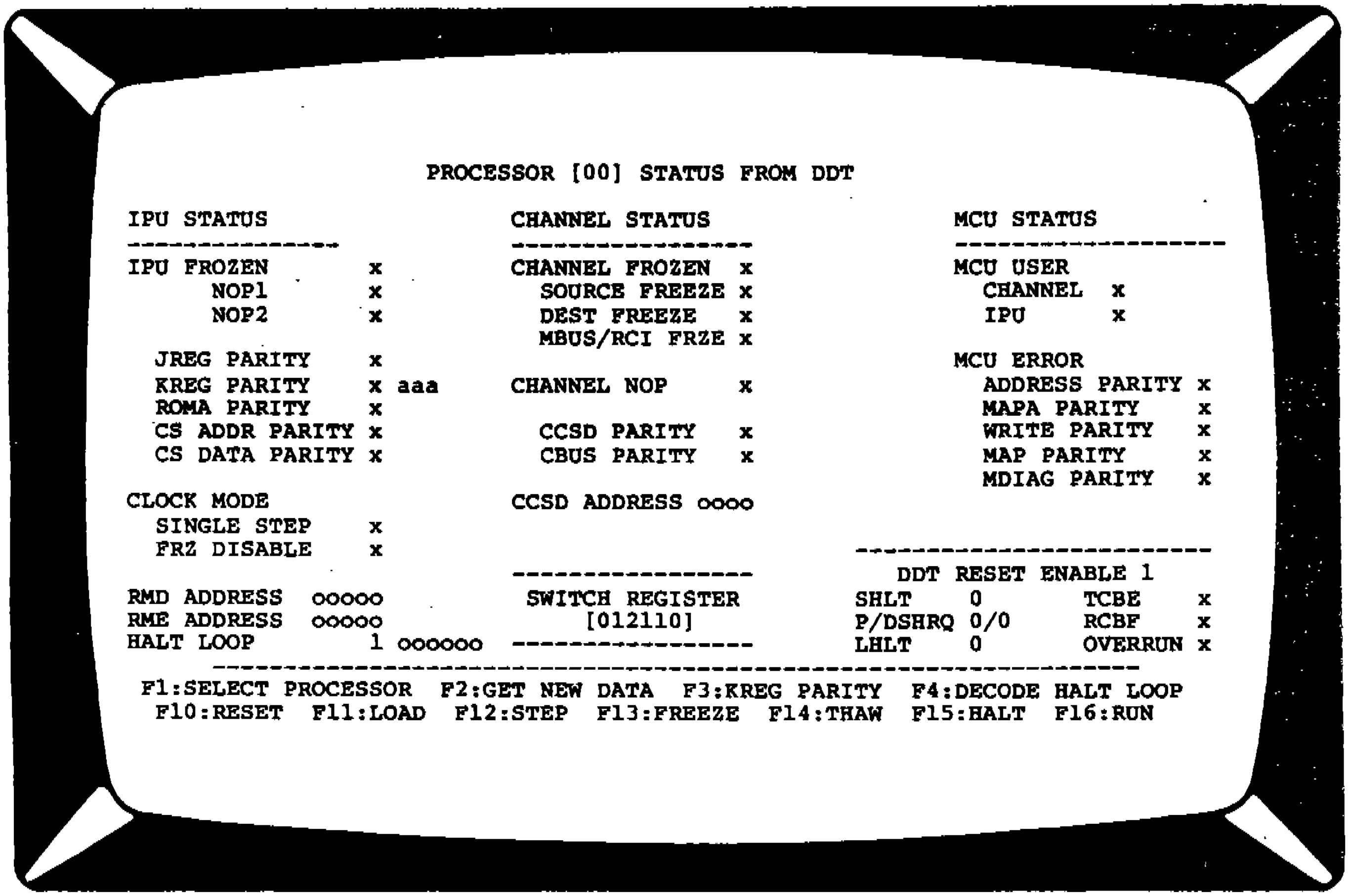
i. Set the date and time of day for the newly loaded system:

```
:SETTIME { <day> <month name> } <year> , <hour>:<minute>
         { <month name> <day> }
```

For example:

```
:SETTIME 20 JULY 1969, 13:20
```

j. The selected processor is now running under GUARDIAN. Execute the procedure described in paragraph 2.2 to reload each of the other processors on the system over the Interprocessor Bus.



```
                      PROCESSOR [00] STATUS FROM DDT

IPU STATUS                CHANNEL STATUS               MCU STATUS
---------------           -----------------            -------------------
IPU FROZEN       x        CHANNEL FROZEN  x            MCU USER
      NOP1       x          SOURCE FREEZE x              CHANNEL  x
      NOP2       x          DEST FREEZE   x              IPU      x
                            MBUS/RCI FRZE x
  JREG PARITY    x                                     MCU ERROR
  KREG PARITY    x aaa    CHANNEL NOP     x              ADDRESS PARITY x
  ROMA PARITY    x                                       MAPA PARITY    x
  CS ADDR PARITY x          CCSD PARITY   x              WRITE PARITY   x
  CS DATA PARITY x          CBUS PARITY   x              MAP PARITY     x
                                                         MDIAG PARITY   x
CLOCK MODE                CCSD ADDRESS oooo
  SINGLE STEP    x
  FRZ DISABLE    x                                    ------------------------
                          -----------------             DDT RESET ENABLE 1
RMD ADDRESS  ooooo          SWITCH REGISTER           SHLT     0     TCBE    x
RME ADDRESS  ooooo             [012110]               P/DSHRQ 0/0    RCBF    x
HALT LOOP        1 oooooo -----------------           LHLT     0     OVERRUN x
---------------------------------------------------------------------------
 F1:SELECT PROCESSOR  F2:GET NEW DATA  F3:KREG PARITY  F4:DECODE HALT LOOP
  F10:RESET  F11:LOAD  F12:STEP  F13:FREEZE  F14:THAW  F15:HALT  F16:RUN
```

T16/8123-01

Figure 2-1 PROCESSOR STATUS FROM DDT Screen After Cold Load

2.4 ERROR INDICATIONS WHEN COLD LOAD FAILS

When a cold load completes successfully, the message PROCESSOR UP is displayed at the OSP terminal and the value %177777 is indicated in the processor Switch Register lights (but not in the SWITCH REGISTER field of the PROCESSOR STATUS FROM DDT screen).

Failure to complete a cold load operation results in the display of other codes in the Switch Register lights. These codes are listed in Table 2.1. Two specific error conditions cause messages to be display also at the OSP terminal. These are:

a. FAILURE TO CREATE INITIAL C.I. NEWPROCESSERR <error no.>

The system cannot create a start-up Command Interpreter. <error no.> denotes the File System error, which may help in determining the corrective action to be taken.

b. UNABLE TO OPEN OSIMAGE <error no.>

The system cannot load the Operating System Image because the the file in which it resides cannot be opened. <error no.> denotes the File System error, which may help in determining the corrective action to be taken.

Should one of these messages be displayed, the system delays for a time to allow observation of the message; then the system freezes.

The error codes listed in Table 2.1 give a rough indication of the error types that may be detected during a cold load. If a more exact specification of the error is needed, refer to the extended discussion of cold load errors in the OSP User Guide, Part Number 82801.

Table 2.1 Cold Load Errors Displayed in the Switch Register Lights

OCTAL CODE (%)	DESCRIPTION
20	SCHANL error
21	Uncorrectable Memory Error
22	Memory access breakpoint
23	Instruction failure
24	Page Fault error
30	CCL condition on an EIO instruction
31	CCG condition in an IIO instruction
32	Bad INTERRUPT CAUSE word
33	Bad IOC STATUS word
34	Checksum error on disk read
35	Tape still not readable after 10 retries
36	Tape in wrong format
37	Interrupt timeout
40	LCS did not verify
41	Bad SWITCH REGISTER option
42	Disk bootstrap not found in volume label
43	Memory size in error (insufficient memory available for requested configuration)
44	Directory not found in volume label
45	System Image file not found
46	File error in directory
47	Not enough physical memory

PART 6

Using EXERCISE

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 2, PART 6

USING EXERCISE

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

December 1982
Tandem Part Number 82803
Printed in U.S.A.

VOLUME 1, CHAPTER 2, PART 6

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION..1-1

SECTION 2
RUN PROCEDURE

2.0 RUN PROCEDURE...2-1

LIST OF TABLES

Table No. Title Page No.

2.1 Checklist for Running EXERCISE....................2-6

PART 6 - USING EXERCISE
TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The Tandem NonStop (TM) system provides both for diagnostics that run in an offline environment and for diagnostics that run in an online environment. Those that run in an offline environment are, for the most part, written in the SHADOW diagnostic language and run under the SHADOW operating system. The procedures for loading SHADOW and other standalone diagnostics are described in Volume 1, Chapter 2, Part 1, of this manual. Those that run in an online environment are written in the EXERCISE diagnostic language and run under the GUARDIAN operating system. The procedure for invoking EXERCISE is described in Section 2.

SECTION 2
RUN PROCEDURE

2.0 RUN PROCEDURE

Since EXERCISE is a GUARDIAN process, running EXERCISE is a matter simply of locating the volume and subvolume where the EXERCISE code resides, and then of executing the Command Interpreter command to run the program. While this procedure is relatively simple, certain ramifications of diagnostic execution can be complicated. To avoid them, the run environment must meet certain specifications before the running of EXERCISE is attempted. Otherwise, depending on the actual EXERCISE facilities invoked, problems may be encountered. To execute EIO instructions, for instance, EXERCISE absolutely requires that the process XOSINTRF be present in $SYSTEM.SYSTEM. Checkout of the environment consists chiefly of confirming the presence and accessibility of this and other program files necessary for EXERCISE. These files are listed in Table 2.1.

The procedure for running EXERCISE is as follows:

a. Locate a terminal running a Command Interpreter process.

b. Ensure that the process $SYSTEM.SYSTEM.XOSINTRF exists on the system and is licensed.

> CAUTION
>
> EXERCISE diagnostic programs cannot execute EIO instructions without the XOSINTRF process. On some systems, the process may still exist under the file name XOSTNS2, the name originally assigned on the SUT tape. If so, the file name must be changed to XOSINTRF. The XOSINTRF process contains privileged code; if it is not licensed when an executing program attempts to access it, the program aborts, and the error message "NEW PROCESS ERROR 7" (unlicensed privileged program) is displayed. Licensing, which enables a user to access a file even though it contains privileged code, requires SUPER.SUPER capability. Licensing is a function of the FUP command LICENSE, described in the GUARDIAN Operating System Command Language and Utilities Manual, Part Number 82073.

c. Enter the following command at the CI prompt (:) and depress carriage return:

```
VOLUME $SYSTEM.EXERCISE
```

NOTE

> The files required for EXERCISE, both object and source code, ordinarily reside in this subvolume. The subvolume can, however, reside in any volume or, if not present on the system, be loaded from tape.

d. Ensure that subvolume $SYSTEM.EXERCISE contains the following files:

1. The object-code file EXERCISE (file type 100).

2. The source-code file EXERLIBS (file type 101). EXERLIBS is required only if any EXERCISE compiles are to be done. Compiles are necessary in the following instances: if object-code files for the EXERCISE diagnostics to be run do not exist on the subvolume; if these diagnostics are modified before being run; or if original EXERCISE programs are to be written and then run. These situations rarely occur, and the general user is not advised to attempt a compile procedure unless it is absolutely necessary.

3. The source-code files (file type 101) and corresponding object-code file (file type 100) and message file (file type 410) for the individual EXERCISE diagnostics to be run. A typical source-code file is DISCEX02 (Rev level E04); the corresponding object-code and message files are DP2VE04 and DM2VE04, respectively.

NOTE

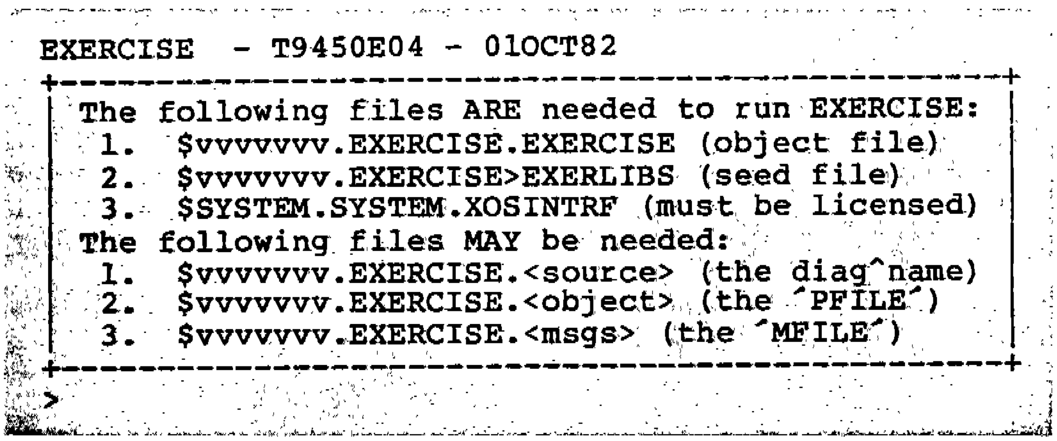
The names of the respective object-code and message files for all EXERCISE programs are derived in a standard fashion: the three (sometimes four) digits preceding the V identify the EXERCISE source program and distinguish the object-code file from the message file (P for object-code and M for message); the V itself stands for "version;" and the last three digits reflect the current revision level (E04, for example). The object-code and message files for TAPEX03 are, for instance, TXP3VA01 and TXM3VA01. The names and types of the files on a given subvolume may be obtained with the FUP command INFO. Refer to the GUARDIAN Operating System Command Language and Utilities Manual, Part Number 82073, for a description of this command. Any of the required files not present on the EXERCISE subvolume may be loaded from tape.

e. Enter at the CI prompt the following command and depress carriage return:

```
RUN EXERCISE
```

NOTE

When the EXERCISE process is started, the following banner and cautionary note is displayed together with the EXERCISE prompt (>):



```
EXERCISE  - T9450E04 - 01OCT82
+---------------------------------------------------+
|The following files ARE needed to run EXERCISE:    |
|  1.  $vvvvvvv.EXERCISE.EXERCISE (object file)     |
|  2.  $vvvvvvv.EXERCISE>EXERLIBS (seed file)       |
|  3.  $SYSTEM.SYSTEM.XOSINTRF (must be licensed)   |
|The following files MAY be needed:                 |
|  1.  $vvvvvvv.EXERCISE.<source> (the diag^name)   |
|  2.  $vvvvvvv.EXERCISE.<object> (the ´PFILE´)     |
|  3.  $vvvvvvv.EXERCISE.<msgs> (the ´MFILE´)       |
+---------------------------------------------------+
>
```

f. At the EXERCISE prompt (>), enter the following command to take off line the device to be tested (unless it is already off line):

```
DOWN <device name>
```

NOTE

The DOWN command is a function of the EXERCISE command set. It is passed on to PUP and is used to make the device inaccessible to user processes. The PUP command DOWN may be used directly to accomplish the same end, as in some cases will already have been done before EXERCISE is entered. This is usually the case when the device is known to be malfunctioning or if it has been physically removed from an unreconfigured system. If PUP DOWN has already been issued for the device to be tested, EXERCISE DOWN should be omitted. Refer to the NonStop II System Operations Manual, Part Number 82075, for a description of the PUP command DOWN. (PUP may require SUPER.GROUP capability.)

g. When the EXERCISE prompt is redisplayed, enter the following command and depress carriage return:

```
RUN <filename>
```

where <filename> is the source-code file for any EXERCISE diagnostic on the EXERCISE subvolume, such as DISCEX02.

CAUTION

When EXERCISE receives instruction to run a given diagnostic, it first creates a temporary copy of the source-code file for that program, named ZZEX###. If changes are made to the diagnostic before it is run, EXERCISE records those changes in the temporary copy of the source-code file, not the original. Since changes or attempted changes to the source-code file make the original object-code file obsolete, EXERCISE purges the original object-code file and attempts to create a new one by recompiling the temporary source-code file. For the compile operation, EXERLIBS is required. If it is not present, EXERCISE is forced to abort. Unlike SHADOW compiles, which complete in several seconds, EXERCISE compiles can take forty minutes or more. Because of these possible complications, the general user is advised not to attempt any procedures involving an EXERCISE compile. For a description of the run procedure for given diagnostic programs, refer to the appropriate discussions elsewhere in this manual.

Table 2.1 Checklist for Running EXERCISE

REQUIRED FILE	CONSIDERATIONS
$SYSTEM.SYSTEM.XOSINTRF	Object-code file (file type 100). Contains privileged code; must be licensed. EXERCISE diagnostics cannot execute device EIOs without XOSINTRF.
EXERCISE.EXERCISE	Object-code file (file type 100). Ordinarily found on $SYSTEM. If not permanently in place, can be loaded from tape. This file is the EXERCISE program.
EXERCISE.EXERLIBS	Source-code file (file type 101). Contains standard TAL routines for EXERCISE. Required if EXERCISE is to perform any compile operations. If not permanently in place on the EXERCISE subvolume, can be loaded from tape. Compiles are typically performed in the following contexts: a. If original EXERCISE programs are to be written and run b. If existing EXERCISE programs are to be modified before running c. If object-code files do not already exist for specified source-code files.
Source-code files for individual EXERCISE diagnostics	(File type 101). If not permanently in place on the EXERCISE subvolume, can be loaded from tape.
Object-code files for individual EXERCISE diagnostics	(File type 100). If not permanently in place on the EXERCISE subvolume, can be loaded from tape.
Message files for individual EXERCISE diagnostics	(File type 410). If not permanently in place on the EXERCISE subvolume, can be loaded from tape.

CHAPTER 3

Diagnostic Utility Programs

PART I

DIAGBOOT

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 1, CHAPTER 3, PART 1

DIAGBOOT

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

REVISION RECORD

REVISION	DATE	DESCRIPTION
A	Aug. 1981	First Edition
B	Dec. 1982	1) Manual updated to reflect software changes through 1 OCT 82 release. 2) Editorial changes.

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 1, CHAPTER 3, PART 1

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1
INTRODUCTION

1.0 INTRODUCTION..1-1
1.1 INVOKING DIAGBOOT..1-2

SECTION 2
CREATING A DIAGNOSTIC COLD LOAD TAPE

2.0 CREATING A DIAGNOSTIC COLD LOAD TAPE.....................2-1
2.1 PRERUN CONSIDERATIONS...2-2
2.2 TRANSPARENT USE OF DIAGBOOT TO CREATE A COLD LOAD TAPE...2-3
2.3 INTERACTIVE USE OF DIAGBOOT TO CREATE A COLD LOAD TAPE...2-6

SECTION 3
UPDATING OR LISTING FILES ON A DIAGNOSTIC COLD LOAD TAPE

3.0 UPDATING OR LISTING FILES ON A DIAGNOSTIC COLD LOAD TAPE..3-1
3.1 UPDATING OR LISTING FILES INTERACTIVELY.................3-1
3.2 UPDATING OR LISTING FILES TRANSPARENTLY.................3-4

SECTION 4
LOADING A TEST PROCESSOR ONLINE FROM TAPE OR DISC

4.0 LOADING A TEST PROCESSOR ONLINE FROM TAPE OR DISC........4-1

SECTION 5
ERROR CONDITIONS

5.0 ERROR CONDITIONS...5-1

LIST OF ILLUSTRATIONS

Fig. No.	Title	Page No.
2-1	Structure of EDIT File MAKECOLD	2-4
3-1	Sample Command File for Updating a Cold Load Tape	3-4
4-1	System Resources for Online Load from Tape or Disc	4-2
5-1	Tape Controller Status Formats - T16/3202	5-3
5-2	Tape Controller Status Formats - T16/3206	5-4

LIST OF TABLES

Table No.	Title	Page No.
1.1	Examples of Legal Syntax for the DIAGBOOT Command	1-4
4.1	Standard SHADOW Switch Register Settings	4-9
4.2	Sample Responses to NEXT FILE? Prompt During Online Load from Tape or Disc	4-10
5.1	Error Conditions During Diagnostic Load	5-2

B00

PART 1 - DIAGBOOT
TABLE OF CONTENTS

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

The diagnostic programs needed for hardware testing in the field are typically carried by the customer engineer on a diagnostic cold load tape, or DCT. Although the programs are sometimes available on disc at the customer site, it is advisable to have a DCT because disc files may not always be accessible.

DIAGBOOT creates a diagnostic cold load tape by writing a series of bootstrap loaders at the beginning of a tape, followed by some specified set of diagnostic program files. In addition, DIAGBOOT may be used to add new diagnostic programs to an existing DCT, to substitute new versions of diagnostic programs for old versions already on the tape, or simply to list the file names of the programs on the tape. Finally, DIAGBOOT must be invoked explicitly when loading a test processor online from tape or disc.

To summarize, then, DIAGBOOT is a diagnostic utility program with the following primary functions:

a. Creating a diagnostic cold load tape (DCT)

b. Updating an existing DCT

c. Listing the file names of the programs on an existing DCT

d. Loading SHADOW or other standalone diagnostics online from tape or disc.

The use of DIAGBOOT to create a diagnostic cold load tape is described in Section 2. The use of DIAGBOOT to update or list the files on an existing tape is described in Section 3. The use of DIAGBOOT to load SHADOW or other standalone diagnostics online from tape or disc is described in Section 4. (A description of this last use of DIAGBOOT can also be found in Volume 1, Chapter 2, Part 1, of this manual, "Loading SHADOW and Other Standalone Diagnotics.") Refer to paragraph 1.1 for a detailed description of the syntax for the command to run DIAGBOOT.

PART 1 - DIAGBOOT
INTRODUCTION

1.1 INVOKING DIAGBOOT

One of the following two forms of the Command Interpreter (CI) command to run DIAGBOOT is the appropriate syntax, depending on the location of the object code for DIAGBOOT. If, as is ordinarily the case, the object code for DIAGBOOT resides in $SYSTEM.SYSTEM, the following command may be entered regardless of the user's current volume and subvolume:

```
      DIAGBOOT / IN <command file>,OUT <list file> / <tape file>
```

If the object code for DIAGBOOT resides in some other volume and subvolume than $SYSTEM.SYSTEM, the following command syntax must be used and may only be entered from the volume and subvolume in which the DIAGBOOT code is located:

```
    RUN DIAGBOOT / IN <command file>,OUT <list file> / <tape file>
```

In either case, the elements of the command syntax are interpreted as follows:

<command file> specifies a file from which responses to the DIAGBOOT inquiries are to be read. This is typically either the home terminal or an EDIT-type file, the successive lines of which must contain valid responses (one per line) to the successive inquiries displayed by DIAGBOOT as it runs. If an EDIT file is specified, DIAGBOOT in effect runs transparent to the user, obtaining the required responses, line by line, from the EDIT file. If no <command file> is specified, this parameter defaults to the home terminal. In this case, DIAGBOOT runs interactively, obtaining the required responses from discrete user inputs at each prompt.

<list file> specifies a file to receive all listing output from DIAGBOOT. This is typically either a line printer or the home terminal. If a line printer is specified, all output from DIAGBOOT is sent to that destination. This option is typically chosen when DIAGBOOT is being used simply to list the files on an existing tape and a hard-copy listing is desired. If no <list file> is specified, this parameter defaults to the home terminal.

<tape file> specifies the device name of the tape unit on which the cold load tape is mounted. This parameter is omitted only when DIAGBOOT is being used to load diagnostics into a test processor online from disc. (Refer to Section 4.)

As this explanation of parameters suggests, several truncated forms of the DIAGBOOT command are accepted as legal syntax and may be appropriate depending on the way DIAGBOOT is to be used. Some examples of legal syntax and of the DIAGBOOT environment invoked by each are listed in Table 1.1.

Table 1.1 Examples of Legal Syntax for the DIAGBOOT Command

COMMAND FORM	ENVIRONMENT INVOKED
DIAGBOOT/IN MAKECOLD,OUT $LP/$TAPE1	DIAGBOOT runs transparent to the user, taking responses from MAKECOLD, copying from disc the diagnostics named in MAKECOLD, placing them on the tape mounted on $TAPE1, and printing output at the device named $LP.
DIAGBOOT/IN MAKECOLD/$TAPE1	As in the example above, DIAGBOOT runs transparent to the user, taking responses from MAKECOLD, copying from disc the diagnostics named in MAKECOLD, and placing them on the tape mounted on $TAPE1. The only difference is that it prints the output at the home terminal.
DIAGBOOT $TAPE1	DIAGBOOT runs interactively at the home terminal. This form of the command is used for either of the following two cases: a. When loading a test processor online from tape b. When creating or updating a diagnostic cold load tape
DIAGBOOT	DIAGBOOT runs interactively at the home terminal. This form of the command is used only when loading a test processor online from disc.

SECTION 2
CREATING A DIAGNOSTIC COLD LOAD TAPE

2.0 CREATING A DIAGNOSTIC COLD LOAD TAPE

When creating a diagnostic cold load tape, the user may specify the desired files in one of the following two ways:

a. As a list in an EDIT-type file, which is in turn passed as the IN parameter of the DIAGBOOT command line. Subvolume ZZDIAG contains a file called MAKECOLD that is customarily used for this purpose. MAKECOLD lists all of the diagnostic programs for the NonStop and NonStop II systems (taken from subvolumes ZZDIAG, ZZDIAG1, and ZZDIAG2). When passed as the IN parameter in a DIAGBOOT command, MAKECOLD causes a standard diagnostic cold load tape to be created. This method is described in paragraph 2.2.

b. One by one at a terminal running a DIAGBOOT process, as a succession of responses to the DIAGBOOT prompt NEXT FILE?. This method is described in paragraph 2.3.

When the first method is used, DIAGBOOT in effect runs transparent to the user; when the second method is used, DIAGBOOT runs interactively. Unless a DCT with a very specific (and limited) number of files is desired, the user ordinarily chooses the first method to create a cold load tape. In both instances, the files are copied from existing disc files, typically located in three subvolumes, ZZDIAG, ZZDIAG1, and ZZDIAG2. The volume location of these subvolumes varies from system to system. ZZDIAG contains the diagnostic programs shared by both the NonStop and the NonStop II systems; ZZDIAG1 contains those that apply only to the NonStop system; and ZZDIAG2 those that apply only to the NonStop II system.

The standard DCT contains all offline, standalone diagnostics for the NonStop and NonStop II systems, including SHADOW and the peripheral diagnostics written in SHADOW. In addition to the diagnostics themselves, the DCT always contains the bootstrap loaders necessary for cold loading a test processor on either system.

Certain prerun considerations are discussed in paragraph 2.1.

2.1 PRERUN CONSIDERATIONS

Before attempting to create a diagnostic cold load tape (DCT), the following steps should be taken:

a. Ensure that the system on which the DCT is to be made has the current software releases of DIAGBOOT and of the diagnostic programs to be placed on the tape.

NOTE

> The DIAGBOOT program is ordinarily located in $SYSTEM.SYSTEM.DIAGBOOT. The complete set of diagnostic programs are ordinarily contained in three subvolumes, ZZDIAG, ZZDIAG1, and ZZDIAG2, the volume location of which varies from system to system. Since diagnostic cold load tapes are not typically made on the systems where they will ultimately be used, the question of revision levels and software compatibility can come into play. For instance, the version of DIAGBOOT used to create the cold load tape determines the version of the bootstrap loader placed on the tape. If this version or the versions of the diagnostic programs themselves are not compatible with the revision level of a system where use of the DCT is later attempted, the diagnostic load procedure may abort or suspend abnormally. In general, to avoid such problems, diagnostic cold load tapes should always be made (or updated) with current software releases.

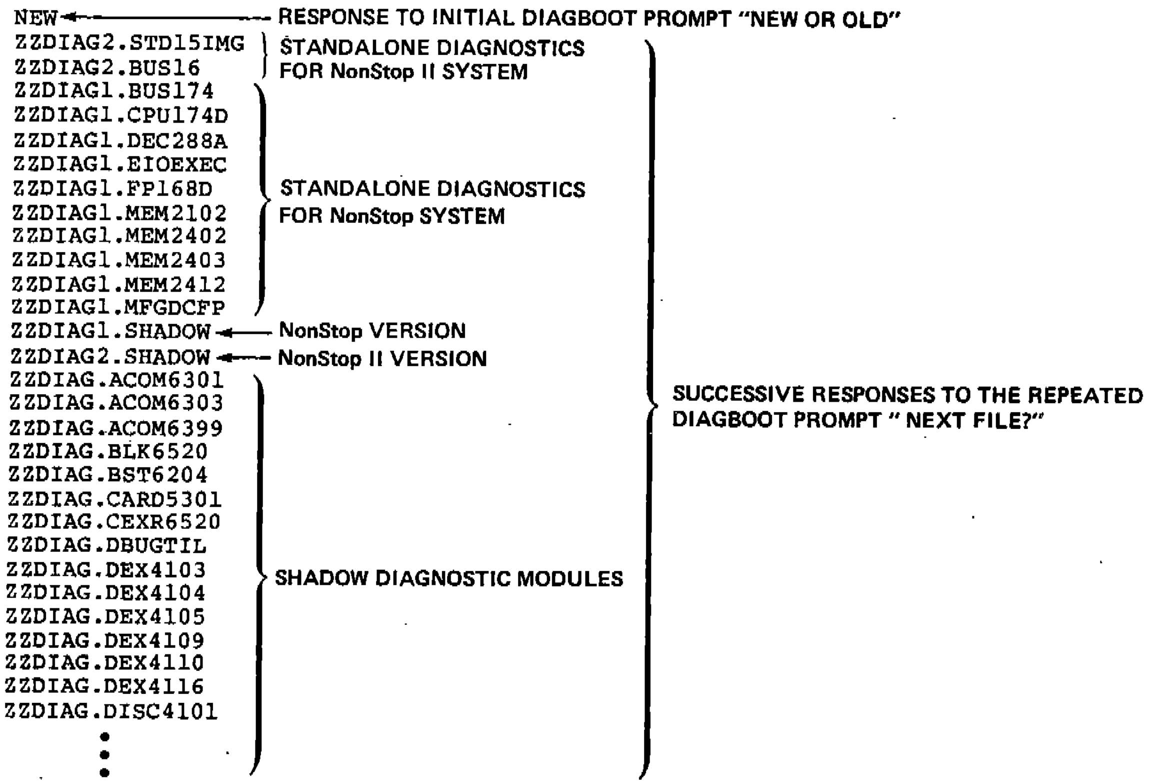
b. Mount the diagnostic cold load tape on an operational tape drive, ensuring that the density setting is correct, that the tape advances properly to the load point, and that the unit comes on line.

2.2 TRANSPARENT USE OF DIAGBOOT TO CREATE A COLD LOAD TAPE

DIAGBOOT can create a diagnostic cold load tape while running transparent to the user. To do this, the user must have available an EDIT-type file that contains a line by line list of responses to the DIAGBOOT prompts, including the file names of the programs to be loaded on the tape. Only one file name may be specified per line, and the file names must be sufficiently qualified to specify the location of the files on disc (volume and subvolume if different from the current). This file is then passed as the IN parameter of the DIAGBOOT command line.

The first line of this file must contain the single specification "NEW". This corresponds to the initial response required by DIAGBOOT when running interactively. (In interactive mode, DIAGBOOT first prompts for the user to indicate whether the tape is NEW OR OLD. If the tape is new, DIAGBOOT automatically loads the bootstrap loader at the beginning of the tape.) The subsequent lines of the file act as responses to the successive NEXT FILE? prompts that DIAGBOOT would issue if it were running interactively. DIAGBOOT terminates when it reads the end of file or a blank line.

Ordinarily, the user creating a diagnostic cold load tape in this way simply uses MAKECOLD, the file designed for this purpose and located on subvolume ZZDIAG. The user may, of course, create an IN file tailored to a particular application and containing any subset of the files listed in MAKECOLD. This file may bear any valid file name. The only requirement is that the general structure of the MAKECOLD file be retained; the first line of the file must specify NEW, and subsequent lines must contain single file names sufficiently qualified for location on disc. To illustrate this general structure, the first thirty lines of ZZDIAG.MAKECOLD are shown in Figure 2-1 (with annotation).



```
NEW
ZZDIAG2.STD15IMG
ZZDIAG2.BUS16
ZZDIAG1.BUS174
ZZDIAG1.CPU174D
ZZDIAG1.DEC288A
ZZDIAG1.EIOEXEC
ZZDIAG1.FP168D
ZZDIAG1.MEM2102
ZZDIAG1.MEM2402
ZZDIAG1.MEM2403
ZZDIAG1.MEM2412
ZZDIAG1.MFGDCFP
ZZDIAG1.SHADOW
ZZDIAG2.SHADOW
ZZDIAG.ACOM6301
ZZDIAG.ACOM6303
ZZDIAG.ACOM6399
ZZDIAG.BLK6520
ZZDIAG.BST6204
ZZDIAG.CARD5301
ZZDIAG.CEXR6520
ZZDIAG.DBUGTIL
ZZDIAG.DEX4103
ZZDIAG.DEX4104
ZZDIAG.DEX4105
ZZDIAG.DEX4109
ZZDIAG.DEX4110
ZZDIAG.DEX4116
ZZDIAG.DISC4101
      •
      •
      •
```

RESPONSE TO INITIAL DIAGBOOT PROMPT "NEW OR OLD"
STANDALONE DIAGNOSTICS FOR NonStop II SYSTEM
STANDALONE DIAGNOSTICS FOR NonStop SYSTEM
NonStop VERSION
NonStop II VERSION
SHADOW DIAGNOSTIC MODULES
SUCCESSIVE RESPONSES TO THE REPEATED DIAGBOOT PROMPT " NEXT FILE?"

T16/8131-01

Figure 2-1 Structure of the EDIT file MAKECOLD

The procedure for creating a diagnostic cold load tape in this way is as follows:

a. Observe the prerun considerations described in paragraph 2.1. If unfamiliar with the DIAGBOOT syntax or with the various valid forms of the DIAGBOOT command, refer to paragraph 1.1.

b. At a terminal with a Command Interpreter (CI), logon at the CI prompt (:), access the volume and subvolume containing the IN parameter file MAKECOLD, and execute one of the following two forms of the command to run DIAGBOOT:

```
DIAGBOOT / IN MAKECOLD, OUT <list file> / <tape file>
```

or

```
DIAGBOOT / IN MAKECOLD / <tape file>
```

where <list file> is the name of the device to receive a listing of the loaded files and <tape file> is the device name of the tape drive unit, such as $TAPE or $TAPE1. If the second form (omitting <list file>) is used, the listing occurs at the home terminal.

NOTE

This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to paragraph 1.1 for the appropriate syntax.

2.3 INTERACTIVE USE OF DIAGBOOT TO CREATE A DIAGNOSTIC COLD LOAD TAPE

In addition to the transparent mode described in paragraph 2.2, DIAGBOOT may also be used interactively to create a diagnostic cold load tape. This procedure is as follows:

a. Observe the prerun considerations described in paragraph 2.1. If unfamiliar with the DIAGBOOT syntax or with the various valid forms of the DIAGBOOT command, refer to paragraph 1.1.

b. At a terminal with a Command Interpreter (CI), logon at the CI prompt (:), and execute the command to run DIAGBOOT:

```
DIAGBOOT <tape file>
```

where <tape file> is the device name of the tape drive unit, such as $TAPE or $TAPE1. This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to paragraph 1.1 for the appropriate syntax.

c. When the following prompt is displayed:

```
NEW OR OLD?
```

enter the letter N (NEW) and depress carriage return.

d. When the following prompt is displayed at the CI terminal:

```
NEXT FILE?
```

enter the file name of the diagnostic program to be written on the tape, such as ZZDIAG2.SHADOW or ZZDIAG.DEX4110, and depress carriage return. The specified file names must be sufficiently qualified to be located on disc (volume and subvolume when different from current). DIAGBOOT simply writes the specified program on the tape and reissues the NEXT FILE? prompt, requesting the file name of the next program to be written. This process is repeated until the user terminates the tape creation by entering a simple carriage return without other input at the NEXT FILE? prompt.

CAUTION

Two distinct versions of SHADOW exist (both called simply SHADOW). One is found in ZZDIAG1.SHADOW and is for use on NonStop systems; the other is found in ZZDIAG2.SHADOW and is for use on NonStop II systems. If the DCT being created may later be used on both types of systems, both versions of SHADOW should be written on the tape. (When the DCT is later used to load a test processor, the bootstrap loaders, which are capable of identifying the system type at run time, automatically select and load the appropriate version of SHADOW.)

CREATING A DIAGNOSTIC COLD LOAD TAPE

SECTION 3
UPDATING OR LISTING FILES ON A DIAGNOSTIC COLD LOAD TAPE

3.0 UPDATING OR LISTING FILES ON A DIAGNOSTIC COLD LOAD TAPE

When updating or listing files on an existing cold load tape, the user ordinarily invokes DIAGBOOT interactively. The transparent use of DIAGBOOT is also, of course, possible, assuming the availability of an appropriate EDIT file to be passed as the IN parameter in the command to run DIAGBOOT.

The interactive use of DIAGBOOT to update or list the files on an existing cold load tape is described in paragraph 3.1. The transparent use is described in paragraph 3.2.

3.1 UPDATING OR LISTING FILES INTERACTIVELY

The interactive procedure for updating or listing the files on an existing tape is as follows:

a. Observe the prerun considerations described in paragraph 2.1. If unfamiliar with the DIAGBOOT syntax or with the various valid forms of the DIAGBOOT command, refer to paragraph 1.1.

b. At a terminal with a Command Interpreter (CI), logon at the CI prompt (:), and execute the command to run DIAGBOOT:

```
DIAGBOOT <tape file>
```

where <tape file> is the device name of the tape drive unit, such as $TAPE or $TAPE1. This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to paragraph 1.1 for the appropriate syntax.

c. When the following prompt is displayed:

```
NEW OR OLD?
```

enter the letter O (OLD) at the prompt and depress carriage return.

d. When the following prompt is displayed:

```
LIST FILES ON TAPE?
```

1. If a list of the files on the tape is desired, enter Y at the prompt and depress carriage return.
2. If a list of the files on the tape is not required, enter N at the prompt and depress carriage return.

e. When the following prompt is displayed at the CI terminal:

```
NEXT FILE?
```

enter the file name of the diagnostic program to be written on the tape, such as ZZDIAG2.SHADOW or ZZDIAG.DEX4110, and depress carriage return. The specified file names must be sufficiently qualified to be located on disc (volume and subvolume when different from current). If the specified file does not already exist on the tape, DIAGBOOT simply writes the specified program at the end of the tape and reissues the NEXT FILE? prompt. If, however, the specified file name matches that of a file already on the tape, DIAGBOOT displays the following prompt:

```
FILE ALREADY EXISTS -- DO YOU WANT TO PURGE IT?
```

If the user enters **Y** and carriage return, DIAGBOOT purges the existing file before writing the new file at the end of the tape. Any other response in effect cancels the user´s initial request: DIAGBOOT does not write a new file on the tape; the original version of the file remains, unpurged. DIAGBOOT then prompts for the next file to be written, and the process continues as before, until receipt of carriage return alone terminates the tape update.

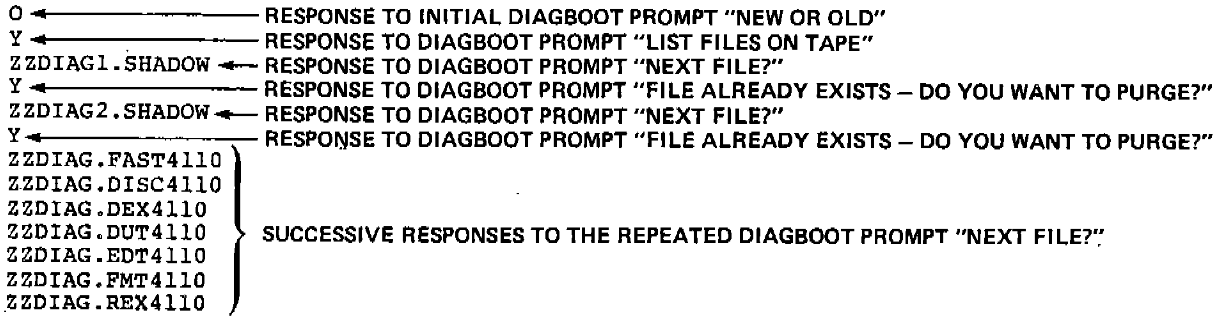
CAUTION

Two distinct versions of SHADOW exist (both called simply SHADOW). One is found in ZZDIAG1.SHADOW and is for use on NonStop systems; the other is found in ZZDIAG2.SHADOW and is for use on NonStop II systems. If the DCT being updated may later be used on both types of systems, both versions of SHADOW should be updated. (When the DCT is later used to load a test processor, the bootstrap loaders, which are capable of identifying the system type at run time, automatically select and load the appropriate version of SHADOW.)

3.2 UPDATING OR LISTING FILES TRANSPARENTLY

The user who is sufficiently familiar with the DIAGBOOT syntax may create an EDIT-type file containing the responses required by DIAGBOOT to implement a particular application, such as listing and then updating certain files on an existing tape. This EDIT file is then passed as the IN parameter of the command to run DIAGBOOT. The only requirement for such a file is that it retain the general structure of the <command file> described in paragraph 1.1. The successive lines of the EDIT file must contain valid responses to the corresponding DIAGBOOT prompts as they would appear if DIAGBOOT were running interactively. DIAGBOOT terminates when it reads the end of file or a blank line. The procedure is analogous to that described in paragraph 2.2 for creating a cold load tape.

The contents of a sample <command file> designed to list the files on an existing tape, to update the existing versions of SHADOW on the tape, and to add the diagnostics for the T16/4110 disc drive, are shown in Figure 3-1 (with annotation).



```
O
Y
ZZDIAG1.SHADOW
Y
ZZDIAG2.SHADOW
Y
ZZDIAG.FAST4110
ZZDIAG.DISC4110
ZZDIAG.DEX4110
ZZDIAG.DUT4110
ZZDIAG.EDT4110
ZZDIAG.FMT4110
ZZDIAG.REX4110
```

RESPONSE TO INITIAL DIAGBOOT PROMPT "NEW OR OLD"
RESPONSE TO DIAGBOOT PROMPT "LIST FILES ON TAPE"
RESPONSE TO DIAGBOOT PROMPT "NEXT FILE?"
RESPONSE TO DIAGBOOT PROMPT "FILE ALREADY EXISTS – DO YOU WANT TO PURGE?"
RESPONSE TO DIAGBOOT PROMPT "NEXT FILE?"
RESPONSE TO DIAGBOOT PROMPT "FILE ALREADY EXISTS – DO YOU WANT TO PURGE?"
SUCCESSIVE RESPONSES TO THE REPEATED DIAGBOOT PROMPT "NEXT FILE?"
T16/8131-02

Figure 3-1 Sample Command File for Updating a Cold Load Tape

SECTION 4
LOADING A TEST PROCESSOR ONLINE FROM TAPE OR DISC

4.0 LOADING A TEST PROCESSOR ONLINE FROM TAPE OR DISC

NOTE

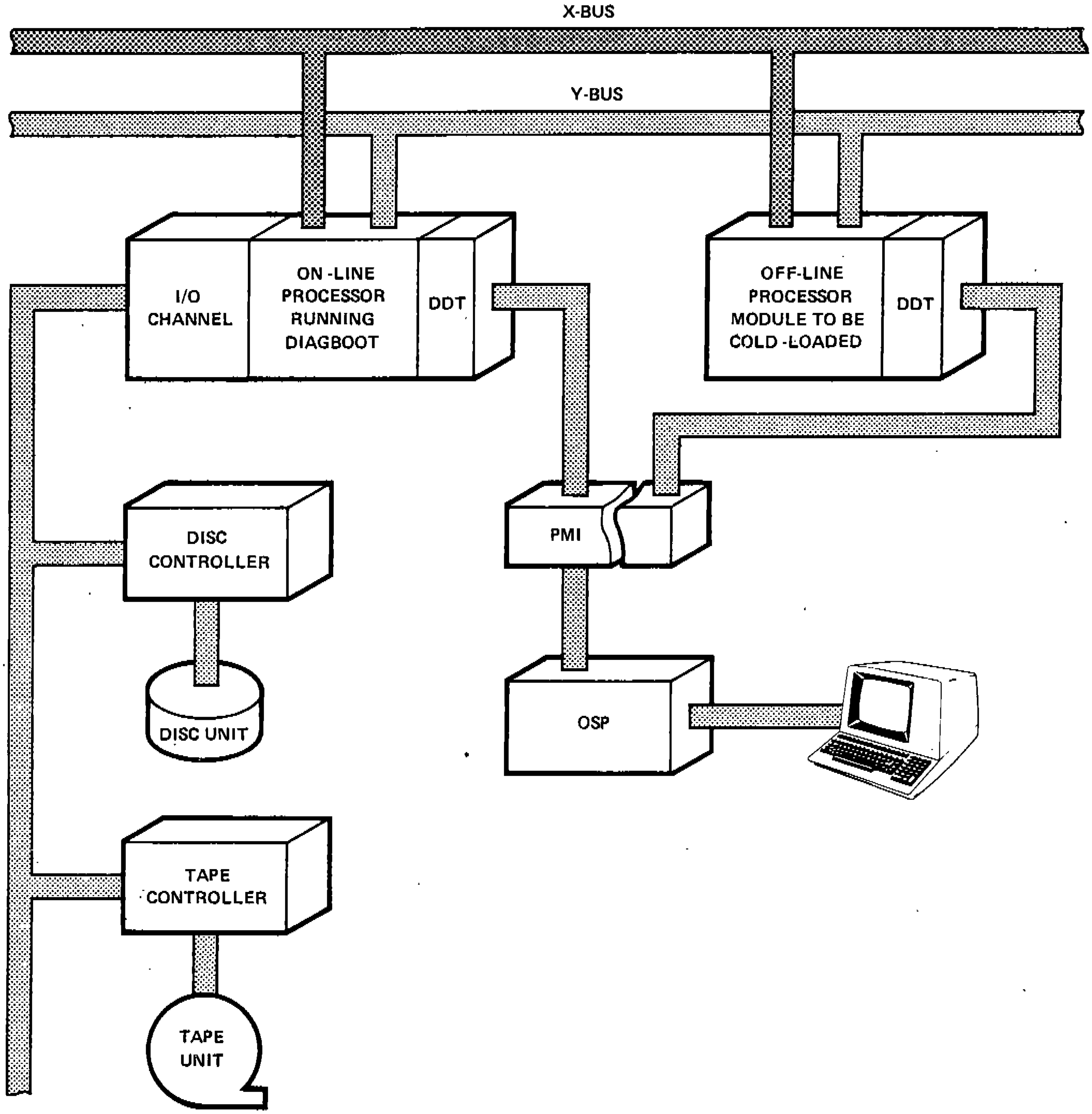
> The following description can also be found in Volume 1, Chapter 2, Part 1, of this manual, "Loading SHADOW and Other Standalone Diagnostics."

When loading a test processor online from tape or disc, DIAGBOOT must be invoked explicitly. For DIAGBOOT to be used for this purpose, the user must have access to two processors connected by the interprocessor bus. The processor to be loaded as the test processor must be off line; the other must be running under the GUARDIAN operating system.

The processor running GUARDIAN must, in turn, have access to the object code for DIAGBOOT and SHADOW and to the source code for the individual diagnostic programs to be run, either on disc or on a diagnostic cold load tape; it must also have access to the OSP subsystem. The offline processor need only have access to the OSP subsystem. The minimal system configuration necessary for online load from tape or disc is illustrated in Figure 4-1.

The only complication of online load is the necessity of coordinating inputs on two different screens of the OSP so that the proper sequence is maintained between sending and receiving processors.

The online load procedures from tape and from disc differ in several minor respects. Principally, when the load is from tape, DIAGBOOT issues several preliminary prompts for input, which must be serviced before it issues the prompt for a file to be loaded. When the load is from disc, there are no preliminary prompts; the initial prompt is for a file to be loaded.



T16/8131-03

Figure 4-1 System Resources for Online Load from Tape or Disc

LOADING A TEST PROCESSOR ONLINE FROM TAPE OR DISC

Before an individual diagnostic program can be selected from GUARDIAN disc files or from a cold load tape, the processor to be loaded must be readied and the DIAGBOOT program must be run in the online processor.

> CAUTION
>
> Before attempting a diagnostic load, disable all processor PMI FREEZE ENABLE switches.

The step-by-step procedure for online load from tape or disc is as follows:

a. Depending on whether the online load is to be from tape or from disc, perform one of the following two steps:

   1. If the load is to be from tape, mount the diagnostic cold load tape on a tape drive unit accessible to the processor running GUARDIAN, ensuring that the density setting is correct, that the tape advances properly to the load point, and that the unit comes on line.

   2. If the load is from disc, ensure that the disc volume containing the object code for DIAGBOOT and SHADOW and the source code for the individual diagnostic programs to be run is accessible to the processor running GUARDIAN.

b. Set the OSP switches to the following positions:

   1. NORMAL/DIAGNOSE to DIAGNOSE

   2. LOCAL/REMOTE to LOCAL

   3. LOCKED/MAINTENANCE to MAINTENANCE

c. Set the PMI switches for the processor to be loaded to the following positions:

   1. DDT ENABLE to ENABLE

   2. FREEZE ENABLE TO DISABLE

   3. RESET ENABLE TO ENABLE

d. At the OSP terminal, depress function key F6 (shifted) to call up the PROCESSOR STATUS FROM DDT screen. When this screen is displayed, proceed as follows:

   1. Use the TAB key to place the cursor over the Processor Number field, enter (in octal) the number of the processor to be loaded with the diagnostic, and depress function key F1 to transmit the processor number.

   2. Depress function key F10 to RESET the processor to be loaded with the diagnostic.

   3. Use the TAB key to place the cursor over the SWITCH REGISTER field, then set bit <0> to 1, and bits <1> through <15> to 0.

   4. Depress function key F11 to ready the processor for a diagnostic load.

e. Depress function key F1 (shifted) to enter Conversational Mode 1 (A Command Interpeter process is now communicating with the OSP terminal.)

f. At the Command Interpreter (CI) prompt (:) execute one of the following two forms of the command to run DIAGBOOT, depending on whether the load is from tape or disc:

   1. If the online load is from tape, enter the following form of the command at the CI prompt and depress carriage return:

```
DIAGBOOT <device name>
```

   where <device name> is the device name of the tape drive unit, such as $TAPE or $TAPE1. This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to Volume 1, Chapter 3, Part 1, of this manual (paragraph 1.1) for the appropriate syntax.

Respond as follows to the several preliminary requests for input:

a) When the following prompt is displayed:

```
NEW OR OLD?
```

enter the letter O (OLD), indicating that the tape already contains the required diagnostic files, and depress carriage return.

NOTE

The other valid response, N (NEW), is appropriate only if a diagnostic cold load tape is being created. (That procedure is described in Section 2.

b) When the following prompt is displayed:

```
LIST FILES ON TAPE?
```

1) If a list of the files on the tape is desired, enter Y at the prompt and depress carriage return.

2) If a list of the files on the tape is not required, enter N at the prompt and depress carriage return.

2. If the online load is from disc, enter the following form of the command at the CI prompt and depress carriage return:

```
DIAGBOOT
```

NOTE

> This syntax assumes that the object code for DIAGBOOT is located in $SYSTEM.SYSTEM. If not, refer to paragraph 1.1 for the appropriate syntax.

g. When the following prompt is displayed at the OSP terminal (Conversational Mode 1),

```
NEXT FILE?
```

enter the file name of the standalone diagnostic to be loaded, such as SHADOW or BUS16, the CPU number of the processor to be loaded, and the interprocessor bus (X or Y) to be used. Then depress carriage return. Refer to Table 4.2 for examples of this input. The syntax of the user input is as follows:

```
<filename>,<cpu><bus>
```

CAUTION

> When loading from disc, it is necessary to specify the exact system location (volume, subvolume, and file) of the standalone diagnostic to be loaded. All diagnostics for the NonStop system and peripherals are customarily located in the subvolumes designated ZZDIAG or ZZDIAG2. The volume specification, however, may vary. If SHADOW diagnostics are to be run, the SHADOW operating system must be loaded before any individual SHADOW diagnostic may be specified. In this case, SHADOW is the only appropriate response to the first NEXT FILE? prompt (not, for instance, DEX4110 or TAPE3206).

h. When the response to NEXT FILE? is received, DIAGBOOT loads the specified standalone into the test processor, and then displays the following prompt:

```
MICROCODE FILE?
```

Proceed as follows:

1. Respond to the prompt by entering the microcode file name ZZDIAG2.STD15IMG and depressing carriage return.

2. When the DIAGBOOT prompt MICROCODE FILE? is redisplayed, simply depress carriage return. (Ordinarily the only microcode file required is STD15IMG.)

3. When the DIAGBOOT prompt NEXT FILE? is redisplayed, do not respond to it directly. Instead depress function key F2 (shifted) to enter Converstional Mode 2.

i. At the OSP terminal (Conversational Mode 2), wait for the banner of the specified standalone to be displayed. (If an error occurs at the receiving processor, the processor halts with an error number displayed in the Switch Register lights. Refer to Section 5 for a discusssion of error conditions in DIAGBOOT.)

When SHADOW is the selected program, the form of the banner is as follows:

```
T9404E00.SHADOW  01APR82  00:00
>
```

NOTE

The exact form of this program banner is release-specific. The SHADOW prompt (>), which immediately follows the banner, indicates that SHADOW is in conversational mode, waiting for valid user input.

j. When the SHADOW prompt (>) is displayed at the OSP terminal (Conversational Mode 2), proceed as follows:

   1. At the operator control panel of the test processor, set bit <0> of the Switch Register to 0.

   2. At the operator control panel of the test processor, set the appropriate bits of the Switch Register to the ON position for the desired SHADOW run-control options. The standard Switch Register settings implemented by SHADOW are listed in Table 4.1. Additional switch settings may apply, depending on the specific SHADOW diagnostic program to be run. For the Switch Register options defined for specific SHADOW diagnostics, refer to the descriptions of those diagnostics.

   3. Enter the following command at the SHADOW prompt:

      ```
      RUN <filename>
      ```

      where <filename> is the name of any available SHADOW diagnostic program, such as DEX4110, POLL6202, or TAPE3206.

   4. Depress carriage return.

k. Depress function key F1 (shifted) to return to Conversational Mode 1, where the DIAGBOOT process is still running. Proceed as follows:

   1. At the the DIAGBOOT prompt NEXT FILE? enter the same file name as that entered in the SHADOW RUN command (Converstional Mode 2), followed by a comma, the CPU number of the processor to be loaded, and the interprocessor bus to be used. The CPU number and bus must be same as those specified when the SHADOW operating system was loaded initially. Refer to Table 4.2 for examples of this procedure.

   2. Depress carriage return.

l. Depress function key F2 (shifted) to return to Conversational Mode 2, where the SHADOW operating system is still running. Enter the simple command RUN and depress carriage return.

NOTE

This sequence must be strictly observed so that the receiving processor is set up to receive the specified program before the sending processor is instructed to send it. For a detailed description of the run procedure for the selected diagnostic program, refer to the appropriate discussion elsewhere in this manual.

Table 4.1 Standard SHADOW Switch Register Settings

SWITCH *	FUNCTION
<0>	If toggled on and off, generates a user break request and returns the SHADOW interpreter to conversational mode, where it waits for further instruction.
<3>	If placed in the ON position, suppresses error printout and enables setting of the special numerics ERROR# and COMPERR#. Refer to the Diagnostic Languages Manual, Part Number 82848, for a description of the special numerics ERROR# and COMPERR#.
<4>	If placed in the ON position, suppresses error printout, disables setting of the special numerics ERROR# and COMPERR#, and disables ERROR limit decrement.
<5>	If placed in the ON position, causes a program halt on I/O error.

* Switches <1>, <2>, <6>, and <7> may be defined for specific diagnostics; switches <8> through <15> reflect the I/O address.

Table 4.2 Sample Responses to NEXT FILE? Prompt During Online Load from Tape or Disc

DIAGBOOT PROMPT	RESPONSE *	RESULT
NEXT FILE?	BUS16,0X <cr>	The program BUS16 is loaded into processor 0 over the X bus, and run.
NEXT FILE?	SHADOW,3Y <cr>	The SHADOW program is loaded into processor 3 over the Y bus, and SHADOW enters conversational mode. (If the load is from tape, the tape rewinds.)
NEXT FILE?	DEX4110,3Y <cr>	Valid response only if SHADOW and microcode file STD15IMG are already loaded in CPU 3 and DEX4110 is also specified at the SHADOW prompt (Conversational Mode 2). If these conditions are met, DEX4110 is loaded into CPU 3 over the Y bus.

* If an asterisk (*) is entered following the name of the diagnostic program to be loaded, the program does not begin execution but halts with %177777 displayed in the Switch Register lights. This option is used mainly for software debugging. To begin program execution, depress function key F16 (shifted) at the PROCESSOR STATUS FROM DDT screen for processor being loaded, or toggle the PROGRAM STEP/RUN switch to RUN at the operator control panel of the processor being loaded.

SECTION 5
ERROR CONDITIONS

5.0 ERROR CONDITIONS

If a serious file system error occurs when DIAGBOOT is running online under GUARDIAN, the process terminates and an appropriate GUARDIAN operating system error message is displayed. For a list of these messages, the conditions they indicate, and the corrective action (if any) to be taken, refer to the GUARDIAN Operating System Messages Manual, Part Number 82076. Other self-explanatory messages may be displayed as DIAGBOOT runs; these are chiefly of the kind originating from operator entry error, which may or may not cause the program to abort.

If an error occurs during an offline load procedure (involving a diagnostic cold load tape created by DIAGBOOT), the processor being loaded halts. A binary error code identifying the condition causing the halt is displayed in the Switch Register lights of the processor. These error codes are listed (in octal) in Table 5.1. They fall into two groups: the first group (negative numbers) occur during the initial phase of the bootstrap load; the second group (positive numbers) occur at any other time during the bootstrap load. In general, errors during the initial phase suggest a malfunctioning tape drive subsystem. Errors after the initial phase are less easily assigned a probable cause. The tape drive subsystem itself is obviously less suspect since it functioned without error during the initial phase. Errors %1 through %7 may indicate a defective tape, while errors %13 and %14 suggest a malfunctioning terminal. Errors %10, %12, and %110 are unlikely occurrences at a customer site; they suggest a program logic error, which may require debugging. The best corrective action in most cases is a simple retry of the load procedure, from a different tape drive or terminal if the error condition suggests it, or with an alternate tape if available. If the halt reoccurs, the error code should be noted and the incident reported.

When a halt is caused by the failure of an I/O operation, certain status information is saved in specific internal registers of the processor. If an EIO instruction fails, the device status is saved in register 0. If an IIO instruction fails, the interrupt cause is saved in register 0; the interrupt status, in register 1. The formats of the EIO and IIO status and cause words for the T16/3202 Tape Controller are shown in Figure 5-1; those for the T16/3206 Tape Controller, in Figure 5-2.

Table 5.1 Error Conditions During Diagnostic Load

SWITCH REGISTER (octal)	ERROR CONDITION
%177776 (-2)	Condition code (CC) after EIO was CCL or CCG
%177775 (-3)	IIO error
%177773 (-5)	Condition code (CC) after IIO was CCG
%177772 (-6)	Ten (10) successive IIO errors on attempt to read record
%177771 (-7)	Attempt to backspace for retry had CCL or CCG after IIO
%177767 (-9)	Initial cold load EIO had CCL or CCG
%000001	Condition code (CC) after EIO was CCL or CCG
%000003	Attempt to backspace for retry had CCL or CCG after EIO
%000004	Attempt to backspace for retry had CCL or CCG after IIO
%000005	Condition code (CC) after IIO was CCG
%000006	10 successive IIO errors on attempt to read record
%000007	Attempt to rewind had CCL or CCG after IIO
%000010	Unexpected interrupt
%000011	Uncorrectable memory error
%000012	Not enough memory to load program
%000013	EIO error on status or write to terminal
%000014	EIO error on read to terminal
%000033	Initial loader tried to load NonStop II System tape bootstrap, but loaded NonStop System tape bootstrap instead
%000070	Too many characters (>28) entered on terminal
%100000 *	Ownership error
%040000 *	Interrupt pending
%020000 *	Channel abort
%010000 *	Parity error

* The interrrupt status errors (also decodable from bits <0> through <3> of the Interrupt Status Word) may occur in combination. The range of possible register displays for these errors is therefore %100000 to %170000. The occurrence of one or more interrupt status errors suggests a malfunctioning tape controller.

EIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	B	P	OV	DPE	CRC	IL	WPV	CR	DNR	TB	EOF	EOT	WR	BOT

O	OWNERSHIP ERROR	WPV	WRITE PROTECT VIOLATION
I	INTERRUPT PENDING	CR	COMMAND REJECT
B	BUSY	DNR	DRIVE NOT READY
P	CHANNEL PARITY ERROR	TB	TAPE BUSY
OV	OVERRUN	EOF	END OF FILE
DPE	DATA PARITY ERROR	EOT	END OF TAPE
CRC	CRC ERROR	WR	WRITE RING
IL	INCORRECT LENGTH	BOT	BEGINNING OF TAPE

Interrupt Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	A	P	0	0	0	0	CONTROLLER NUMBER					UNIT NUMBER		

O	OWNERSHIP ERROR
I	INTERRUPT PENDING
A	ABORT
P	CHANNEL PARITY ERROR

Interrupt Cause Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PON	ATT	RT	UE	OV	DPE	CRC	IL	WPV	CR	DNR	TB	EOF	EOT	WR	BOT

PON	POWER ON INTERRUPT	WPV	WRITE PROTECT VIOLATION
ATT	ATTENTION INTERRUPT	CR	COMMAND REJECT
RT	RUNAWAY TAPE	DNR	DRIVE NOT READY
UE	UNUSUAL END	TB	TAPE BUSY
OV	OVERRUN	EOF	END OF FILE
DPE	DATA PARITY ERROR	EOT	END OF TAPE
CRC	CRC ERROR	WR	WRITE RING
IL	INCORRECT LENGTH	BOT	BEGINNING OF TAPE

T16/8131-05

Figure 5-1 Tape Controller Status Formats - T16/3202

EIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	B	P	U											

O = OWNERSHIP ERROR
I = INTERRUPT PENDING
B = CONTROL UNIT BUSY
P = I/O BUS PARITY ERROR
U = WCS UNLOADED

IIO Status Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
O	I	A	P	RESERVED				CONTROLLER NUMBER					DRIVE NUMBER		

O = OWNERSHIP ERROR
I = INTERRUPT PENDING
A = CHANNEL ABORT
P = PARITY ERROR

T16/8131-06

Figure 5-2 Tape Controller Status Formats - T16/3206

IIO Cause Word:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PON	BOT	EOF	EOT	ON LINE	WRITE ENABLED	REW	RESERVED		TERM-STAT BITS						

PON = POWER ON
BOT = BEGINNING OF TAPE
EOF = END OF FILE
EOT = END OF TAPE
OL = ON LINE
WE = WRITE ENABLED
R = REWINDING

TERMINATION STATUS (bits <9> through <15>):

35	INCORRECT LENGTH	73	REGISTER ERROR
36	WRITE RETRIED	74	BUFFER ERROR
37	READ RETRIED	75	COUNTER/TIMER ERROR
38	SKIP RESIDUE	86	Z80 FAILURE
39	UNCORRECTABLE DATA, CORRECTED	87	PARITY FAILURE
40	UNCORRECTABLE DATA ERROR	88	WRITE-READ LOOP FAILURE
41	FORMATTER COMMAND REJECT	89	REGISTER FAILURE
42	OPERATION ERROR	90	BUFFER FAILURE
43	WRITE FAILURE	91	COUNTER/TIMER FAILURE
44	UNDEFINED COMMAND	102	Z80 PARITY FAILURE
45	BAD MICROCODE FILE	103	CTL FREEZE TIMEOUT
46	LARGE READ ERROR	104	SHORT WRITE
47	RUNAWAY TAPE	105	BAD MEMORY ACCESS
48	NOT READY	106	OPERATION TIMEOUT
70	Z80 ERROR	107	OBUS PARITY FAILURE
71	PARITY ERROR	108	FCU ROM PARITY FAILURE
72	WRITE-READ LOOP	109	ADAPTER FAILURE

T16/8131-07

Figure 5-2 Tape Controller Status Formats - T16/3206 (Cont'd)

VOLUME 2

Processor and Bus Diagnostics

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 2

PROCESSOR AND BUS DIAGNOSTICS

(CURRENT)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

December 1982
Printed in U.S.A.

Tandem Part Number 82803

CURRENT TABLE OF CONTENTS

VOLUME 2 — PROCESSOR AND BUS DIAGNOSTICS

CHAPTER 2 — BUS DIAGNOSTICS

PART 1. BUS16...

CHAPTER 2

Bus Diagnostics

PART 1

BUS16

SECOND LEVEL MANUAL

DIAGNOSTIC OPERATING PROCEDURES

NonStop II (TM)

VOLUME 2, CHAPTER 2, PART 1

BUS16

(REVISION B00)

TANDEM COMPUTERS INCORPORATED
2450 Walsh Avenue
Santa Clara, California 95050

Tandem Part Number 82803

December 1982
Printed in U.S.A.

REVISION RECORD

REVISION	DATE	DESCRIPTION
A	Aug. 1981	First Edition
B	Dec. 1982	Editorial changes.

REVISION LETTERS I, O, Q, U AND X ARE NOT USED

VOLUME 2, CHAPTER 2, PART 1

TABLE OF CONTENTS

Par.No. Title Page No.

SECTION 1 INTRODUCTION

1.0 INTRODUCTION..1-1
1.1 REQUIRED HARDWARE..................................1-1
1.2 OTHER REQUIREMENTS.................................1-3

SECTION 2 OVERVIEW

2.0 OVERVIEW..2-1
2.1 OPERATIONAL CHARACTERISTICS.........................2-1
2.1.1 Conversational Mode...............................2-1
2.1.2 Execute Mode......................................2-2
2.1.3 OSP Debugger Mode.................................2-2
2.2 LOAD PROCEDURE......................................2-2
2.3 PROGRAM OPTIONS.....................................2-4
2.3.1 Switch Register Settings..........................2-4
2.3.2 Making a New Master...............................2-5
2.3.3 Testing with Multiple Masters.....................2-6
2.3.3.1 Making Multiple Masters.........................2-6
2.3.3.2 Viewing Errors Logged by a Non-Communicating Master..2-7
2.4 REGISTER DISPLAY....................................2-7

SECTION 3 COMMANDS

3.0 COMMANDS..3-1
3.1 BOTH COMMAND..3-2
3.2 CLOCK COMMAND.......................................3-2
3.3 DISPLAYLOG COMMAND..................................3-3
3.4 ERRORS COMMAND......................................3-3
3.5 HELP COMMAND..3-3

TABLE OF CONTENTS (Cont'd)

Par.No. Title Page No.

3.6 IDENT COMMAND....................................3-4
3.7 LIST COMMAND.....................................3-4
3.8 LOG COMMAND......................................3-5
3.9 MASTER COMMAND...................................3-5
3.10 NOERRORCHECK COMMAND............................3-6
3.11 OTHERCPUS COMMAND...............................3-6
3.12 REINITIALIZE COMMAND............................3-6
3.13 SLAVE COMMAND...................................3-7
3.14 TEST COMMAND....................................3-7
3.15 WRITE COMMAND...................................3-10
3.16 X-BUS COMMAND...................................3-11
3.17 Y-BUS COMMAND...................................3-11
3.18 COMMAND SUMMARY.................................3-12

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS..................4-1
4.1 DIRECT DISPLAY MODE.................................4-1
4.1.1 Faulty Packet Transfers...........................4-2
4.1.2 Failures in Detecting Forced Errors...............4-3
4.1.3 Operator Entry Errors.............................4-5
4.2 MESSAGE LOG MODE....................................4-5
4.3 LOGGING TABLE MODE..................................4-6
4.4 BUS INTERRUPT STATUS CODES..........................4-6
4.5 TIME-TAGGING OF ERROR MESSAGES......................4-7

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURE.......................................5-1
5.1 SUMMARY...5-1

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Fig.No.	Title	Page No.
1-1	BUS16 Test Configuration	1-2
4-1	Bus Interrupt Status Word	4-6

LIST OF TABLES

Table No.	Title	Page No.
2.1	Switch Register Settings	2-5
3.1	TEST Command Test Operations	3-8
3.2	Sample TEST Command Usages	3-9
3.3	BUS16 Commands	3-12
4.1	Bus Interrupt Status Codes	4-7

SECTION 1
INTRODUCTION

1.0 INTRODUCTION

This document describes the Multiple-CPU Interprocessor Bus diagnostic BUS16 (T9471A00). BUS16 is a T/TAL language cold load standalone diagnostic used for verifying proper operation of the Interprocessor Bus of a Tandem NonStop II (TNS II) processing system.

1.1 REQUIRED HARDWARE

BUS16 requires the proper operation of the entire TNS II processor module, including memory and I/O channel logic. (64K bytes of main memory are required.) Decimal and floating-point microcode are not needed. Memory, Clocks and IPB Control (MCB) board bus control logic is part of the hardware tested by the diagnostic and is not required to operate properly for testing to be carried out.

A properly functioning Operations and Service Processor (OSP) must be configured in the TNS II system. Refer to "OSP User Guide" (82801 and "OSP Maintenance Manual" (82846 for detailed information concerning the OSP.

Figure 1-1 shows the standard field test configuration and the hardware tested by the diagnostic.

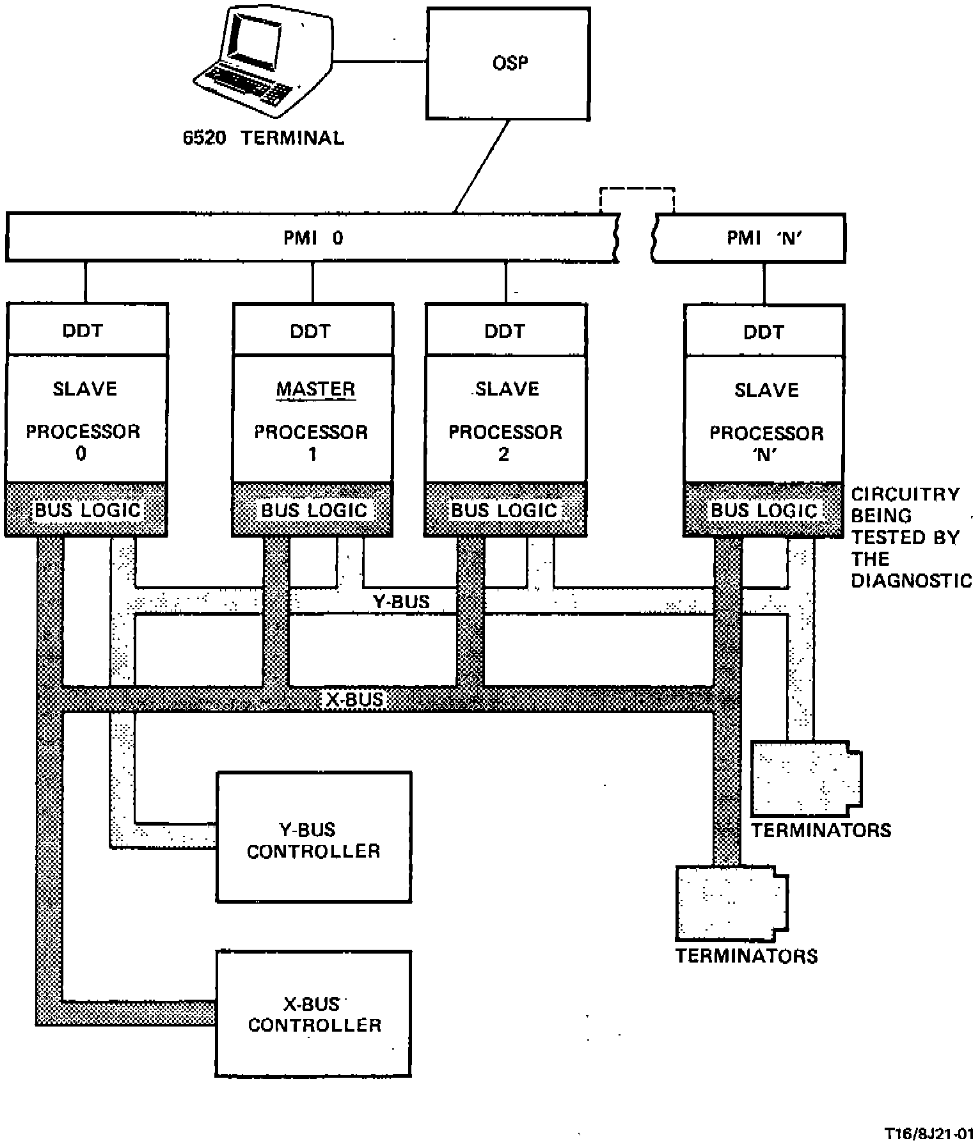


Figure 1-1 BUS16 Test Configuration

1.2 OTHER REQUIREMENTS

All processors to be loaded with the diagnostic must be offline. They must be brought on line when testing has been completed. Refer to Chapters 1.2 and 1.3 for descriptions of the procedures for taking a processor module off line, bringing a processor module on line, and cold loading the system from disc.

The total time needed for running BUS16 is variable. Test operations are controlled by the user after the program is loaded; the longest test operation lasts approximately fifteen seconds.

CAUTION

BUS16 runs as a standalone process in one or more CPUs without interfering with others. However, a CPU which is running GUARDIAN can be stopped if test packets are accidentally sent to it through either incorrect routing by faulty bus logic or the wrong use of the OTHERCPUS or WRITE commands by the operator. The operator must guarantee that no packets are sent to a CPU which is running GUARDIAN and any user applications.

SECTION 2
OVERVIEW

2.0 OVERVIEW

BUS16 tests the operation of the Interprocessor Buses and their associated hardware by sending packets of bus data and verifying their correct transmission. This diagnostic is able to test the entire data path, including from the transmitters to the edge connector (J1), without the aid of an IPB Transponder. To achieve this, one or more "master" CPUs transfer data to other "slave" CPUs acting as "smart" transponders.

Only a master CPU can initiate a test operation, but both master and slave CPUs are capable of detecting and recording bus errors during data transfers. BUS16 test operations consist of a master either sending data to itself (testing its own bus logic) or sending data to a slave. Slave CPUs are required to echo all packets sent them by a master.

BUS16 is a standalone diagnostic which can coexist with as many as fifteen other copies of itself in fifteen other CPUs. Only one master CPU is allowed to communicate with the OSP. The master designation may be shifted to different CPUs during program execution, and multiple masters can be created to carry out simultaneous data transfer tests.

2.1 OPERATIONAL CHARACTERISTICS

BUS16 has the following three modes of operation:

a. Conversational mode

b. Execute mode

c. OSP Debugger mode

2.1.1 Conversational Mode

The program accepts user commands from the OSP in the conversational mode. In this mode, the decimal CPU number of the currently communicating master, followed by a "greater than" sign (">"), is displayed as a prompt for command entry. For example:

```
(15)>
```

User commands are entered (one per line) and followed by a carriage return. The following special characters are recognized:

control "H"	- backspace
control "X" and ESCAPE key	- cancel line
BREAK key	- user break request

Refer to "OSP User Guide" (82801-B00) for a description of the procedures for using all operational modes of the OSP.

2.1.2 Execute Mode

Transfers of data packets as specified by the TEST and WRITE commands are carried out in the execute mode. When a master CPU is in the execute mode, it does not accept input from the OSP (other than the BREAK function). A master CPU in the execute mode (and any slave CPU) can only be controlled using Switch Register switches.

2.1.3 OSP Debugger Mode

The OSP Debugger mode is used strictly for program maintenance and is outside the scope of this manual. For a description of the OSP Debugger facility, refer to "OSP User Guide" (82801-B00).

2.2 LOAD PROCEDURE

Ensure that all processors to be tested are offline. Refer to Chapter 1.2 for a description of the procedure for taking a processor module off line.

Cold load each CPU with the diagnostic. It can be loaded from a cold load tape either directly or over the Interprocessor Bus from an another offline processor. It may also be loaded from tape or GUARDIAN disc file from an online processor by running the program in $SYSTEM.ZZDIAG.DIAGBOOT and entering the tape or disc filename for BUS16 when requested. (The GUARDIAN disc file named $SYSTEM.ZZDIAG.BUS16 should be accessed if performing an online load from disc.) Refer to Chapter 1.1 for a complete description of TNS II diagnostic cold load procedures, and Chapter 3.1 for a description of the DIAGBOOT program.

NOTE

The Switch Register field of the "Processor Status from DDT" screen can not be used during BUS16 program execution to specify or to alter program options. Once a processor under test has been cold loaded with the diagnostic, its Switch Register switches must be set manually.

Each CPU awakes as a master when it is loaded, displaying the program banner and its own CPU number at the OSP terminal. After each CPU is loaded, it must be made a slave before another CPU can be loaded. This is achieved by toggling Switch <2> on and off on the Operator Control Panel or by entering "SLAVE" after the prompt displayed and depressing the carriage return. For example:

```
(04)> SLAVE     <cr>
```

The CPU (04 in this example) then goes into an idle state as a slave, and ceases communication with the OSP. The next CPU loaded awakes as a master, displaying the following prompt (the CPU number is an example only):

```
(03)>
```

All further messages during program execution are preceded by the number of the currently communicating master CPU.

CAUTION

Attempts to load additional CPUs can not succeed if the current master is not first made a slave, since multiple communicating masters would otherwise exist. (The conditions under which multiple masters can exist are limited; refer to paragraph 2.3.3 for their description.)

Refer to Section 5 of this chapter for a summary of the steps necessary to load and run the diagnostic. Refer to Chapters 1.1 and 3.1 of this manual for descriptions, respectively, of TNS II diagnostic cold load procedures and of the DIAGBOOT program.

2.3 PROGRAM OPTIONS

After the diagnostic has been loaded, options may be selected by entering commands at the keyboard and altering Switch Register settings. Options include the abilities to name a new master CPU and to run tests using multiple master CPUs. Refer to Section 3 of this chapter for an explanation of all BUS16 keyboard commands.

2.3.1 Switch Register Settings

The Switch Register switches are used during program execution to specify test options. The Switch Register settings and their functions are shown in Table 2.1.

NOTE

> In Table 2.1, the options associated with switches <3:8> are dynamic, and may be altered during program execution. After the diagnostic is loaded it does not recognize Switch Register changes made at the OSP. Any changes must be made at the Operator Control Panel.

Table 2.1 Switch Register Settings

SWITCH NOS.	SWITCH FUNCTION (UP EQUALS "TRUE")
<12:15>	Not used.
<9:11>	Not used.
<8>	Make all output go to message log and ignore communications from OSP (even if a master).
<7>	Loop on test. (Used for any test, or for WRITE command.)
<6>	Display whole packet after error. Also, cause full data display after a checksum error.
<5>	Halt on error. (Test can be restarted by enabling "RUN" on service panel or at the OSP terminal.)
<4>	Suppress direct display of errors (not error logging).
<3>	Disable time-of-day printouts.
<2>	Set slave mode if toggled on and then off.
<1>	Set master mode if toggled on and then off. Also, sync each packet with the clock (10 msec) if the switch is left on.
<0>	If a master, escape back to conversational mode when done with current test.

2.3.2 Making a New Master

When the current master is in the conversational mode, a new CPU can be designated a master by using the MASTER command. (The old master then becomes a slave.) The testing parameters specified in the following commands are global and are transferred to the new master: BOTH, CLOCK, LOG, OTHERCPUS, X-BUS, and Y-BUS.

The MASTER command aborts if neither bus is capable of transmitting command packets to the designated CPU. Switch Register switches may then be used to designate a new master. Any CPU which is not busy testing can be made a master using Switch <1> or a slave by using Switch <2>. Whenever a CPU changes from slave to master its CPU number becomes part of the prompt displayed at the OSP (for example: "(08)>").

The MASTER command cannot be used if Switch <8> has been toggled on to cause the present master to stop communicating with the OSP. To make a new master, toggle Switch <1> on and off on the CPU that is to become the master. (This condition is normally encountered only when making multiple masters. Refer to paragraph 2.3.3.)

2.3.3 Testing with Multiple Masters

Continuous tests can be run from multiple masters. All masters except the one which communicates with the OSP send error message output to memory. (This is necessary when multiple masters exist because only one master can communicate with the OSP at a time.) Multiple masters may use any other CPUs as slaves which are specified by the OTHERCPUS command and are not themselves functioning as masters. Two or more masters may send packets to the same slave CPU simultaneously.

2.3.3.1 Making Multiple Masters

The following steps are necessary in order to create multiple masters:

a. For the current master, set Switch <7> on to enable looping and start a specific test process executing.

b. Set Switch <8> of the current master on. This causes all messages to be diverted to memory and instructs the CPU to stop communicating with the OSP. The CPU remains a master for testing purposes and continues to execute the specified tests.

c. Make another CPU a master by toggling its Switch <1> on. The MASTER command cannot be used to make this CPU a master, since no CPU is currently able to receive keyboard commands.

NOTE

> No CPU which is acting as a slave for an executing test process should be made a master.

d. Start a test process from the new master using the TEST command.

If another master is desired, Steps b through d are repeated.

2.3.3.2 Viewing Errors Logged by a Non-Communicating Master

To examine the message log (refer to paragraph 4.2) or logging table (refer to paragraph 4.3) of a master that is not currently communicating with the OSP:

a. Ensure that no other master is able to communicate with the OSP (Switch <8> should be on for all masters).

b. Toggle Switch <8> off to make a communicating master of the CPU whose errors are to be examined.

c. Toggle Switch <7> off to cause the current test to stop looping.

d. Use the ERRORS command to display the message log contents or the DISPLAYLOG command to display the logging table contents.

2.4 REGISTER DISPLAY

Bits <8:15> in the Register Display of a master which is in the conversational mode are on. When the master enters the execute mode, the display shows the origin (sending CPU number) of the last-received data packet. The origin of the packet appears as the master itself if a local (self) test was tried, or as a slave echoing a packet if a non-local test was tried. (Tests 3 through 8 are local and the rest are non-local.)

The Register Display of a slave CPU contains the Bus Interrupt Parameter word. (Refer to Figure 4-1.) Bits <8:15> show the origin of the last received packet. The sender is always a master CPU.

SECTION 3
COMMANDS

3.0 COMMANDS

This section describes in detail all BUS16 commands. Refer to paragraph 3.18 for a brief summary, in tabular form, of BUS16 commands.

Commands of the BUS16 program are accepted only by the currently communicating master CPU. (Only one CPU can communicate with the OSP at a time.) After the program has been loaded and one CPU is a communicating master, that master accepts keyboard commands in the conversational mode.

Once a BUS16 test operation is initiated through a TEST command issued to a master, the master is in the execute mode and does not accept user commands until the test is complete. (Switch Register settings are recognized by a master during a TEST command operation.)

When a new master is made, the current parameters specified in the following commands are transferred to that master: BOTH, CLOCK, LOG, OTHERCPUS, X-BUS, and Y-BUS. Refer to paragraph 2.3.2 for a description of the procedure for making a new master, and paragraph 2.3.3 for a description of testing with multiple masters.

NOTE

> After the program has been loaded, all CPUs containing the BUS16 program and available for testing must be specified by the OTHERCPUS command. Any command parameter containing a CPU number unspecified by the OTHERCPUS command is rejected as invalid input.

In the following definitions of BUS16 commands, each parameter is defined when it first occurs. Later occurrences do not include a restatement of definition. Refer to Chapter 1.6 of this manual for an explanation of the metalanguage conventions used.

3.1 BOTH COMMAND

The form of the BOTH command is as follows:

```
both_command  ::=  BOTH
```

The BOTH command enables the testing of both buses if issued prior to the execution of a test. The mode of testing enabled by the BOTH command is the initial, default option. Refer to X-BUS and Y-BUS commands for alternate test modes.

Examples:

```
     (00)>BOTH

     (00)>B
```

3.2 CLOCK COMMAND

The form of the CLOCK command is as follows:

```
clock_command  ::=  CLOCK {hour:minute | 0:0}

  hour  ::=  a two-digit decimal number in the range 00 to 23
  minute  ::=  a two-digit decimal number in the range 00 to 59
```

The CLOCK command is used to enable and disable the time-tagging of error messages.

If "hour:minute" is specified, the CLOCK command sets the time of day and starts the clock of the master. At the same time it synchronizes and starts the clocks of all CPUs specified by the OTHERCPUS command.

If "0:0" is specified, the clock is stopped and reset to zero. "0:0" also stops and resets the clocks of all other CPUs specified by the OTHERCPUS command. When the clock time is set to "0:0", error messages are not accompanied by a time stamp.

Examples:

```
     (10)>CLOCK 12:30

     (10)>C 0:0
```

3.3 DISPLAYLOG COMMAND

The form of the DISPLAYLOG command is as follows:

```
displaylog_command  ::=  DISPLAYLOG
```

The DISPLAYLOG command causes the logging table of the current master to be displayed at the OSP. The logging table of a slave can be examined by designating it the new master (using the MASTER command or Switch <1>) and then issuing the DISPLAYLOG command. Refer to paragraph 4.3 for the format and meanings of the logging table entries.

Examples:

```
     (07)>DISPLAYLOG

     (07)>D
```

3.4 ERRORS COMMAND

The form of the ERRORS command is as follows:

```
errors_command  ::=  ERRORS
```

The ERRORS command causes the complete message log of the current master to be displayed. The message log of a slave can be examined by designating it the new master (using the MASTER command or Switch <1>) and then issuing the ERRORS command. Refer to paragraph 4.2 for the format and meaning of the message log entries.

Examples:

```
     (15)>ERRORS

     (15)>E
```

3.5 HELP COMMAND

The form of the HELP command is as follows:

```
help_command  ::=  HELP
```

The HELP command causes a local halt of the processor which is the currently communicating master, allowing the user to select the OSP Debugger facility. This facility is for program maintenance and is outside the scope of this manual. For a description of the OSP Debugger facility, refer to "OSP User Guide" (82801-B00).

3.6 IDENT COMMAND

The form of the IDENT command is as follows:

```
ident_command  ::=  IDENT cpu_no

  cpu_no  ::=  any valid decimal CPU number in the range 00 to 15
```

The IDENT command reinitializes the number of the current master CPU. This command is for use in the manufacturing environment only, when it may be necessary to check that a CPU is capable of identifying itself by other numbers than its own. It is necessary to relocate the CPU select cables on the IPB controller board and to rewire the processor number backplane wirewrap area when the IDENT command is used.

CAUTION

The IDENT command should NEVER be used in a system where GUARDIAN is running.

Examples:

```
(03)>IDENT 01

(03)>I 01
```

3.7 LIST COMMAND

The form of the LIST command is as follows:

```
list_command  ::=  ?
```

The LIST command displays a summary of all BUS16 commands and their syntax. The listing is in three parts. After each part, the program asks whether the next part is required and displays a prompt to enter either a "Y" (for "yes") or an "N" (for "no").

Example:

```
(00)>?
```

3.8 LOG COMMAND

The form of the LOG command is as follows:

```
log_command  ::=  LOG
```

The LOG command is used to enable the logging table of a master or a slave. For a CPU to respond to the LOG command, it must be the current master. Logging remains enabled after that CPU has been made a slave. Refer to paragraph 4.3 for the format and meaning of logging table entries.

Examples:

```
     (02)>LOG

     (02)>L
```

3.9 MASTER COMMAND

The form of the MASTER command is as follows:

```
master_command  ::=  MASTER cpu_no
```

The MASTER command is used to designate a new master CPU. When the MASTER command is issued, an acknowledgment is returned by the new master and the old master becomes a slave. Any slave CPU which is not part of a currently executing test process can be specified by the MASTER command.

The MASTER command, like all other BUS16 commands, cannot be used if Switch <8> has been toggled on to cause the present master to stop communicating with the OSP. To make a new master when no master is currently communicating with the OSP, toggle Switch <1> on and off on the CPU that is to become a master. Any existing master CPUs remain masters. (These conditions are normally encountered only when making multiple masters. Refer to paragraph 2.3.3.)

The MASTER command is entered following the prompt displayed by the current master. In the following example, CPU 02 is the current master and CPU 05 the desired new master:

Example:

```
     (08)>MASTER 05

     (08)>M 05
```

3.10 NOERRORCHECK COMMAND

The form of the NOERRORCHECK command is as follows:

noerrorcheck_command ::= NOERRORCHECK

The NOERRORCHECK command inhibits all error checking during test execution.

Examples:

```
(15)>NOERRORCHECK

(15)>N
```

3.11 OTHERCPUS COMMAND

The form of the OTHERCPUS command is as follows:

othercpus_command ::= OTHERCPUS [cpu_no...]

The OTHERCPUS command is used to specify all CPUs besides the current master which are connected and available for testing. Only CPUs which are running the BUS16 diagnostic should be specified. Correct use of this command protects against the possibility of test packets being sent to CPUs which are not running BUS16.

If the OTHERCPUS command is issued without parameters, then the current master will be the only CPU available to the program.

Examples:

```
(13)>OTHERCPUS 00,01,02

(13)>O 00,01,02,03,04,05,06,07
```

3.12 REINITIALIZE COMMAND

The form of the REINITIALIZE command is as follows:

reinitialize_command ::= REINITIALIZE

The REINITIALIZE command resets all program flags and options, clears the log area, and causes the program to return to the conversational mode. The CLOCK and OTHERCPUS parameters are retained.

Examples:

```
(07)>REINITIALIZE

(07)>R
```

3.13 SLAVE COMMAND

The form of the SLAVE command is as follows:

```
slave_command  ::=  SLAVE
```

The SLAVE command causes the currently communicating master CPU to become a slave and cease communicating with the OSP. After the SLAVE command is issued from the OSP terminal, no CPU can communicate with the OSP unless it becomes a master as a result of being cold loaded with the BUS16 program, or having Switch <1> toggled on and off at the Operator Control Panel while already a slave.

Examples:

```
(05)>SLAVE

(05)>S
```

3.14 TEST COMMAND

The form of the TEST command is as follows:

```
test_command  ::=  TEST test_op, cpu_no
  test_op  ::=  a decimal number in the range 0 to 9 | @
```

The TEST command causes the program to execute the specified test operation(s). Its format consists of a keyword followed by two parameters. The first parameter specifies the test operation. The second parameter is a slave CPU identifier. The slave CPU number entered must be designated by the OTHERCPUS command as available for testing. If the test operation does not require a slave CPU (Tests 3 through 8), a valid CPU number must still be entered.

The TEST command test operations and their functions are shown in Table 3.1.

Table 3.1 TEST Command Test Operations

TEST NO.	TEST FUNCTION
0	Pattern test over a bus. Thirty-six different patterns are sent at all possible data word counts in the range 1-13.
1	Address test over a bus. A packet is sent containing all possible bit combinations for the CPU address portion of the bus data lines.
2	Not used.
	NOTE Tests 3 through 8 test only the local (master) CPU. The integrity of the bus cable is not tested.
3	Ensure SEND instruction is capable of timing out.
4	Not used.
5	Ensure unsequenced packets are transmitted properly.
6	Send out-of-sequence packet and check termination.
7	Send unexpected packet and check termination.
8	Test TOTQ instruction for proper operation.
9	Alternately send over the X-bus, then the Y-bus. Performs the same pattern test as Operation 0, except that the packets are sent alternately over the X-bus and Y-bus. (The BOTH command must be specified. Refer to paragraph 3.1.)
@	Run all tests.

Table 3.2 shows some examples of TEST commands and their functions.

Table 3.2 Sample TEST Command Usages

COMMAND	COMMAND FUNCTION
TEST @,@	Perform all tests over current buses, with all declared CPUs acting as slaves for Tests 0, 1, and 9.
TEST @,15	Perform all tests over current buses, with CPU 15 acting as slave for Tests 0, 1, and 9.
TEST 0,15	Perform Test 0 over current bus, with CPU 15 acting as slave.

Whenever the execution of all tests has been specified (TEST @,...) and completed, the display shows the CPUs and buses which were tested and the passcount in the following format:

```
PASS=-----  OF BUS16  ON CPUS: ---- ---- ---- ----, BUS: X&Y
```

In the 16-bit representation of the CPUs which were tested, the leftmost digit shows the test state of CPU 00 and the rightmost shows the state of CPU 15. All CPUs loaded with BUS16 (and specified by the OTHERCPUS command) are identified by either an "M" or an "S", meaning master or slave. In the following example, CPUs 00, 03, 05, and 12 are slaves, and CPU 09 is a master; the remaining CPUs have not been specified by the OTHERCPUS command:

```
PASS=07661  OF BUS16  ON CPUS: S--S -S-- -M-- S---, BUS: X&Y
```

3.15 WRITE COMMAND

The form of the WRITE command is as follows:

```
write_command  ::=  WRITE cpu_no, bus_spec, data_pat, pat_len,
                          echo_int

  bus_spec   ::=  X | Y
  data_pat   ::=  an octal number in the range %0-%177777
  pat_len    ::=  a decimal number in the range 1-28
  echo_int   ::=  a decimal value of 40 time increments or higher
                  (suggested value)
```

The WRITE command allows the sending of a packet of a chosen data pattern to a CPU over one of the buses. The length of the pattern in words is specified, as is the waiting time for the echo to be received.

"cpu_no" names the CPU to which the data will be sent.

"bus_spec" designates the bus which will be used.

"pat_len" gives the length of the data pattern in words.

"echo_int" specifies the interval of time allowed for the packet to be sent and returned from the echoing CPU. Time is expressed in increments of 10.0 microseconds.

NOTE

> The WRITE command overrides the effects of the X-BUS, Y-BUS, and OTHERCPUS commands. Care must be taken that data is not accidentally sent to a CPU which is not running BUS16.

Examples:

```
(00)>WRITE 01,X,52525,30,100

(00)>W 13,Y,177777,36,500
```

3.16 X-BUS COMMAND

The form of the X-BUS command is as follows:

```
x-bus_command  ::=  XBUS
```

The X-BUS command enables exclusive testing of the X-bus by causing the Y-bus to be bypassed during test execution. The X-BUS command is cancelled by issuing the BOTH command or the Y-BUS command.

Examples:

```
(06)>XBUS

(06)>X
```

3.17 Y-BUS COMMAND

The form of the Y-BUS command is as follows:

```
y-bus_command  ::=  YBUS
```

The Y-BUS command enables exclusive testing on the Y-bus by causing the X-bus to be bypassed during test execution. The Y-BUS command is cancelled by issuing the BOTH command or the X-BUS command.

Examples:

```
(11)>YBUS

(11)>Y
```

3.18 COMMAND SUMMARY

Table 3.3 contains all BUS16 commands. Each command is accompanied by a brief description.

Table 3.3 BUS16 Commands

COMMAND	DESCRIPTION
BOTH	Enable testing of both buses (default condition).
CLOCK	Start and stop time-tagging of error messages.
DISPLAYLOG	Display logging table contents for current master.
ERRORS	Display message log contents for current master.
HELP	For program debugging only. Causes local halt.
IDENT	Reinitialize master CPU identity.
? ["LIST"]	Display all commands and their syntax.
LOG	Enable logging table of current master.
MASTER	Designate another CPU master.
NOERRORCHECK	Inhibit all error checking during test execution.
OTHERCPUS	Specify CPUs loaded with BUS16 (protective).
REINITIALIZE	Reset flags and options; clear logs; re-enter conversational mode.
SLAVE	Make currently communicating master a slave.
TEST	Execute data packet transfer tests.
WRITE	Send packet; parameters specified by user.
X-BUS	Test X-bus only.
Y-BUS	Test Y-bus only.

SECTION 4
ERROR CONDITIONS AND DISPOSITIONS

4.0 ERROR CONDITIONS AND DISPOSITIONS

Both master and slave CPUs report errors during test execution. Three error reporting modes are implemented:

a. Direct display (to OSP)

b. Message log

c. Logging table

The direct display mode causes the display of error messages at the OSP, and is the default mode of logging for a master CPU. Messages include the identity of the CPU reporting the error and, optionally, the time of day. Refer to paragraph 4.1 for an explanation of direct display mode error messages and their format. Refer to paragraph 4.5 for a description of the time-tagging of messages.

The message log mode is identical to the direct display mode in error message format, but its output is stored in memory. The message log mode is the initial, default mode of message reporting for a slave but can also be used by a master (if its Switch <8> is actuated). Refer to paragraph 4.2 for an explanation of the message log mode.

The logging table accumulates total error counts and is used by both masters and slaves. The logging table is enabled for a specific CPU by issuing the LOG command while that CPU is the current master. Refer to paragraph 4.3 for an explanation of the logging table.

Error data logged to memory by a CPU can only be viewed while that CPU is the currently communicating master. Before the message log or logging table of a slave can be viewed, that CPU must be made a master.

All error checking by the current master is disabled by the NOERRORCHECK command.

4.1 DIRECT DISPLAY MODE

Errors detected by a master CPU are displayed at the OSP in the direct display mode. This mode functions if the NOERRORCHECK command is not enabled and if Switches <4> and <8> are off. (Errors detected by a slave CPU are logged to memory. Refer to paragraph 4.2.)

The following error messages may be displayed during test execution. The first group of errors includes those that occur during actual data transfers over the Interprocessor Bus. The second include failures to detect errors forced by Tests 3 through 8. The latter group tests only the local (master) CPU and does not send data over the bus cable.

4.1.1 Faulty Packet Transfers

The error conditions resulting from faulty packet transfers over a bus are as follows:

a.
```
NO REPLY IN SPECIFIED INTERVAL IN TEST --X WITH CPU --
```

A Bus Receive Interrupt did not occur within the specified interval during a packet transfer by a master CPU. In the display "TEST --X", the two left-hand digits (represented by "--") specify the Test number, and the right-hand digit (represented by "X") specifies the bus used when the error occurred. "CPU --" specifies the slave CPU.

b.
```
OUTQ BUSY (CCL) IN TEST --X WITH CPU --
                      BUS  SEQ#   CPU#S TMEOUT    ADDRESS   LENGTH
INITIAL SEND STACK: -- ------ ------ ------ 100000------  ------
FINAL  SEND STACK:  -- ------ ------ ------ 100000------  ------
```

The OUTQ failed to become empty within the specified time prior to the execution of a SEND instruction, preventing a CPU from sending a packet. Condition code was set to "CCL."

c.
```
WRONG XFER: CPU & BUS ARE --Y, SHOULD BE --X IN TEST --
```

A master CPU expected to receive an echo over a bus from a CPU. The packet that was echoed contained the wrong sender or bus identifiers. The two left-hand digits displayed represent the echoing CPU. The right-hand digit represents the bus.

d.
```
CHECKSUM ERROR IN DATA TRANSFER IN TEST --X WITH CPU --
   BYTE#   ACTUAL DATA (INCL. SEQ# & CHECKSUM)
  -- & --    - --- --- --- --- ---
  -- & --    - --- --- --- --- ---
  -- & --    - --- --- --- --- ---
  -- & --    - --- --- --- --- ---
```

A checksum error is found by a master CPU upon receiving an echoed packet. The number of data words displayed can be from one to thirteen. If Switch <6> is off, only the first line of the display is shown.

e.
```
BAD STATUS RETURNED ON TRANSFER IN TEST --X WITH CPU --
STATUS SHOULD BE 000   IS ---
```

A bad transfer is indicated by the Bus Interrupt Status parameter word in the System Interrupt Vector (SIV) table. The three octal digits displayed represent the left eight bits of the Bus Interrupt Status word (refer to paragraph 4.4). Both masters and slaves can record this error after a transfer.

f.
```
BRT ADDRESS AND COUNT ARE WRONG AFTER TEST --X WITH CPU --
   ARE:        ADDRESS: ------ ------   COUNT: ------
  SHOULD BE:   ADDRESS: ------ ------   COUNT: ------
```

The transfer address or byte count contained in the Bus Receive Table (BRT) entry for the sender is incorrect. The two six-digit octal numbers after "ADDRESS:" represent the 32-bit BRT address pointer (the first and second word of the extended address contained in the BRT entry for the sender CPU). The six-digit octal number following "COUNT:" is the BRT byte count at the time of the error.

g.
```
BYTES         EXPECTED DATA              BITS IN ERROR
-- & --    - --- --- --- --- ---      - --- --- --- --- ---
TRANSFERRING  --  BYTES ON X-BUS, IN TEST --
```

A data word contained in a returned packet is different from that which was sent. The packet length in bytes, bus identity, and test number are specified.

4.1.2 Failures in Detecting Forced Errors

The error conditions which may result from failures to detect errors forced in Tests 3 through 8 are as follows:

a.
```
X-BUS INCAPABLE OF TIMING OUT IN TEST --
```

A SEND instruction was incapable of timing out.

b.
```
BAD STATUS RETURNED BY UNSEQUENCED PACKET IN TEST --
SIV PARAMETER WORD IS - --- --- --- --- ---
```

The sending of an unsequenced packet did not result in the appropriate Bus Interrupt Status code. (Refer to paragraph 4.4.)

c. **BAD STATUS RETURNED BY OUT-OF-SEQUENCE PACKET IN TEST --**
SIV PARAMETER WORD IS - --- --- --- --- ---

The sending of an out-of-sequence packet did not result in the appropriate Bus Interrupt Status code. (Refer to paragraph 4.4.)

d. **STATUS ERROR TRANSFERRING AN UNEXPECTED PACKET IN TEST --**
SIV PARAMETER WORD IS - --- --- --- --- ---

The sending of an unexpected packet did not result in the appropriate Bus Interrupt Status code. (Refer to paragraph 4.4.)

e. **TOTQ INSTR. FAILED TO RETURN A CCE FOR OUTQUE READY**

A TOTQ instruction failed to return a CCE for "outqueue ready".

f. **TOTQ INSTR. FAILED TO RETURN A CCL FOR OUTQUE BUSY**

A TOTQ instruction failed to return a CCL for "outqueue not ready".

g. **UNEXPECTED CMD FROM CPU --**

A master CPU received a command from another CPU.

h. **UNEXPECTED ACK FROM CPU --**

A slave CPU received a command acknowledgment from another CPU.

i. **NO ACK FROM CPU --**

A master did not receive an expected command acknowledgment from a slave.

4.1.3 Operator Entry Errors

The following messages are displayed when the format or syntax of program input is incorrect:

a. **NON-NUMERIC PARAMETER**

An alphabetic character was found where a numeric character was expected.

b. **BAD OCTAL NUMBER**

A required octal parameter contained an 8 or 9, or was larger than %177777.

c. **NUMBER TOO BIG**

A required decimal parameter contained more than six digits.

d. **--- IS TOO HIGH !**

The decimal number used as input for the OTHERCPUS or IDENT commands exceeded the highest possible value (15).

4.2 MESSAGE LOG MODE

The message log mode is used by slaves unless disabled by Switch <4>, and by masters when Switch <8> is activated. Its messages have the same format as the direct display output to the OSP by a master, but are logged to memory. The message log can be viewed by making the CPU a communicating master and issuing the ERRORS command. If the message log fills up during test execution, that CPU blinks its panel lights once for each discarded message (approximately 1000 messages can be stored).

4.3 LOGGING TABLE MODE

The logging table mode is enabled by the LOG command and viewed by issuing the DISPLAYLOG command. (If the the NOERRORCHECK command has been issued, the logging table does not function, since all error checking is then inhibited.) The logging table contains one entry for each type of error, incremented by one for each occurrence. The error types counted by the logging table are the same as those reported by the direct display and message log facilities. The logging table is displayed in the following format:

```
ERRORS REPLY  TMOUT  RESP   CKSUM  STATS  BRT    DATA
X-BUS  -----  -----  -----  -----  -----  -----  -----
Y-BUS  -----  -----  -----  -----  -----  -----  -----

          FAILURES TO DETECT FORCED ERRORS :
ERRORS:TIME   ROUT-  UNSEQ  OUTOF  UNEXP  TOTQ   TOTQ
        OUT   ING    PACKT  SEQ    PACKT  READY NOT RDY
X BUS: -----  -----  -----  -----  -----  -----  -----
Y BUS: -----  -----  -----  -----  -----  -----  -----
TOTAL ERRORS   X-BUS = ------ ,     Y-BUS = ------
```

4.4 BUS INTERRUPT STATUS CODES

The "PARAMETER WORD" displayed in error printouts is the Bus Interrupt Status parameter word in the SIV table at System Data address %1257. Bit <0> signifies the bus used in the transfer. Bits <1:7> contain an error status code which applies to the most recently received packet. (If only "STATUS" is displayed, it is the extracted left 8 bits of the parameter word.) Bits <8:15> contain the sending CPU number. The form of the Bus Interrupt Status Word is shown in Figure 4-1.

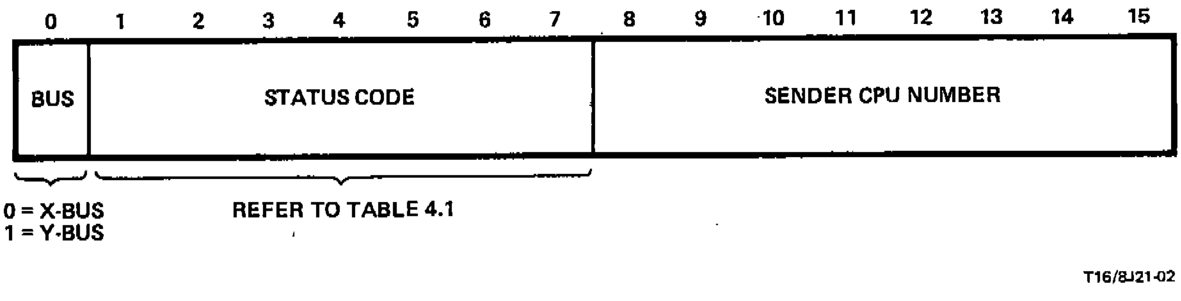


Figure 4-1 Bus Interrupt Status Word

Table 4.1 shows the error status number and meaning for each Bus Interrupt status code (bits <1:7>).

Table 4.1 Bus Interrupt Status Codes

CODE BITS	ERROR STATUS	MEANING
000 0000	0	Normal completion
000 0001	1	Unexpected packet (BRT count was 0)
000 0010	2	Checksum error
000 0011	3	Routing error (inqueue.<0> <> BRT.<3>)
000 0100	4	Unsequenced packet (sequence no. = -1)
000 0101	5	Out-of-sequence packet

The Bus Interrupt parameter word is also displayed in the panel lights during testing. The lights are updated each time there is incoming bus data.

4.5 TIME-TAGGING OF ERROR MESSAGES

A time clock is available for time-tagging all messages. The clock is started by issuing the CLOCK command:

```
CLOCK hh:mm
```

If the clock has been set to a non-zero value and Switch <3> is off, each message from a CPU which is under test shows the time of day. This function allows the correlation of errors when viewing message logs of multiple CPUs.

Refer to paragraph 3.2 for a detailed description of the CLOCK command.

SECTION 5
RUN PROCEDURES

5.0 RUN PROCEDURES

This section contains a summary of the steps necessary to load and run the diagnostic.

> CAUTION
>
> BUS16 runs as a standalone process in one or more CPUs without interfering with others. However, a CPU which is running GUARDIAN can be stopped if test data is accidentally sent to it through incorrect routing of packets by faulty bus logic or by wrong use of the OTHERCPUS or WRITE commands. The operator must guarantee that no packets are sent to a CPU which is running GUARDIAN and any user applications.

5.1 SUMMARY

a. Take off line all processor modules to be loaded with the diagnostic. Refer to Chapter 1.2 for a description of the procedure for taking a processor module off-line.

b. Load the diagnostic program BUS16 into each CPU as required. The GUARDIAN disc file named ZZDIAG.BUS16 should be used if performing an on-line load from disc. Refer to Chapter 1.1 for a description of the standard diagnostic cold load procedures used with the TNS II system. Refer to paragraph 2.2 of this chapter for more detailed information on loading the BUS16 diagnostic.

c. As each CPU is loaded, make it a slave using the SLAVE command. (This is only necessary if another CPU is to be loaded.)

d. After all CPUs are loaded, use the MASTER command to designate a CPU master if desired. (If not done, the CPU loaded last remains the master.)

e. If not already done, select OSP Conversational Mode 2 by depressing Function Key 2 (shifted). Refer to "OSP User Guide" (82801) for a description of the procedures for using all operational modes of the OSP.

NOTE

> If the Switch Register field of the OSP "Processor Status from DDT" screen is used to control the cold load process (rather than the physical switches on the Operator Control Panel), then the OSP Conversational Mode 2 must be selected after a load in order to receive messages from a CPU or to issue BUS16 commands. For example: When loading multiple CPUs, the "Processor Status from DDT" screen is used during each cold load and Conversational Mode 2 is used after each cold load if it is necessary to issue the SLAVE command to the current (last-loaded) master. If Conversational Mode 2 is not selected, the Switch Register can still be used to make the current master a slave.

f. With the master CPU in the conversational mode, use the OTHERCPUS command to designate the CPUs other than the current master which are available for testing purposes. Enter the user commands to specify options (i.e., CLOCK, X-BUS, LOG, etc.).

g. Enter additional run options in the Switch Register switches of the master CPU. (Refer to paragraph 2.3.1.) Further manipulation of Switches <3:8> is possible during actual data transfer tests.

h. Enter commands to initiate the required BUS16 test processes. Refer to Section 3.14 for a detailed description of the TEST command.

NOTE

> If the entire system is offline (no processor is running a program other than BUS16), then no special precautions need be taken prior to entering user commands. If not, the OTHERCPUS and WRITE commands must be used carefully to prevent packets from being sent to the wrong processor(s).

i. After the specified test operations have completed, view the message log of any CPU which sent error messages to memory by making it a master and issuing the ERRORS command. View the logging table (error counts) of any CPU by making it a master and issuing the DISPLAYLOG Command.

j. If desired, designate a new master for further test purposes. (Refer to paragraph 2.3.2.)

k. If desired, initiate simultaneous testing with multiple masters. (Refer to paragraph 2.3.3.)

l. Repeat step i to view error logs. If no further testing is required, terminate program execution by enabling RESET for all processor modules loaded with the diagnostic. The RESET may be initiated at the Operator Service Panel or by using the OSP "Processor Status from DDT" screen RESET function (Function Key F12 [shifted]).